

IBM WebSphere Adapters
Version 7 Release 5

*IBM WebSphere Adapter for Lotus
Domino User Guide
Version 7 Release 5*

IBM

IBM WebSphere Adapters
Version 7 Release 5

*IBM WebSphere Adapter for Lotus
Domino User Guide
Version 7 Release 5*

IBM

Note

Before using this information and the product it supports, read the information in "Notices" on page 155.

November 2011

This edition applies to version 7, Release 5, Fix Pack 1 (7.5.0.1) of IBM WebSphere Adapter for Lotus Domino 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.

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© Copyright IBM Corporation 2006, 2011.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Chapter 1. Overview of WebSphere

| | |
|--|----------|
| Adapter for Lotus Domino | 1 |
| Hardware and software requirements | 2 |
| Technical overview of WebSphere Adapter for Lotus Domino | 2 |
| Outbound processing | 3 |
| Inbound processing. | 10 |
| Business object structures. | 17 |
| External software dependencies. | 20 |
| The external service wizard | 20 |
| Log and Trace Analyzer | 20 |
| Business faults | 21 |
| WebSphere Application Server environment variables | 22 |

Chapter 2. Planning for adapter implementation 25

| | |
|---|----|
| Security | 25 |
| Support for protecting sensitive user data in log and trace files | 25 |
| User authentication. | 26 |
| Deployment options | 26 |
| Migrating to version 7.5 of WebSphere Adapter for Lotus Domino | 30 |
| Migration considerations | 30 |
| Performing the migration. | 31 |
| Upgrading but not migrating a project | 33 |

Chapter 3. Samples and tutorials 35

Chapter 4. Configuring the module for deployment. 37

| | |
|---|----|
| Road map for configuring the module | 37 |
| Creating an authentication alias | 38 |
| Creating the module | 39 |
| Creating the project. | 40 |
| Defining WebSphere Application Server environment variables. | 41 |
| Selecting the processing direction | 44 |
| Setting connection properties for the external service wizard | 45 |
| Configuring the module for outbound processing | 46 |
| Discovering and selecting objects | 46 |
| Configuring the selected objects | 50 |
| Setting deployment properties and generating the service | 51 |
| Configuring the module for inbound processing | 58 |
| Discovering and selecting objects | 58 |
| Configuring the selected objects | 62 |
| Setting deployment properties and generating the service | 63 |

Chapter 5. Modifying artifacts 73

| | |
|-----------------------------------|----|
| Modifying service import. | 73 |
|-----------------------------------|----|

| | |
|------------------------------------|----|
| Modifying service export | 74 |
|------------------------------------|----|

Chapter 6. Changing interaction specification properties 77

Chapter 7. Deploying the module 79

| | |
|---|----|
| Deployment environments | 79 |
| Deploying the module for testing | 79 |
| Generating and wiring a target component for testing inbound processing | 79 |
| Adding the module to the server | 80 |
| Testing the module for outbound processing using the test client. | 81 |
| Deploying the module for production | 81 |
| Installing the RAR file (for modules using stand-alone adapters only) | 82 |
| Exporting the module as an EAR file. | 83 |
| Installing the EAR file. | 84 |

Chapter 8. Administering the adapter module 87

| | |
|--|-----|
| Changing configuration properties for embedded adapters | 87 |
| Setting resource adapter properties for embedded adapters | 87 |
| Setting managed (J2C) connection factory properties for embedded adapters. | 89 |
| Setting activation specification properties for embedded adapters. | 91 |
| Changing configuration properties for stand-alone adapters | 93 |
| Setting resource adapter properties for stand-alone adapters | 93 |
| Setting managed (J2C) connection factory properties for stand-alone adapters | 94 |
| Setting activation specification properties for stand-alone adapters | 95 |
| Starting the application that uses the adapter | 97 |
| Stopping the application that uses the adapter. | 97 |
| Monitoring performance using Performance Monitoring Infrastructure. | 98 |
| Configuring Performance Monitoring Infrastructure. | 98 |
| Viewing performance statistics | 100 |
| Enabling tracing with the Common Event Infrastructure | 101 |

Chapter 9. Troubleshooting and support 103

| | |
|--|-----|
| Configuring logging and tracing | 103 |
| Configuring logging properties | 103 |
| Changing the log and trace file names | 105 |
| Avoiding stale connection problems when invoking the outbound request. | 106 |

| | |
|---|-----|
| Object no longer exists on server exception | 107 |
| org.omg.CORBA.NO_PERMISSION exception. | 107 |
| Server access denied exception | 108 |
| Version conflict exception | 108 |
| Self-help resources. | 109 |

Chapter 10. Reference. 111

| | |
|---|-----|
| Business object information. | 111 |
| Business object attributes for operations | 111 |
| Naming conventions | 114 |
| Fault business objects. | 114 |
| Configuration properties. | 115 |
| Guide to information about properties | 116 |
| Outbound configuration properties | 117 |
| Inbound configuration properties. | 130 |
| Deployment and configuration of the WALD inbound extension manager | 144 |

| | |
|---|-----|
| View and verify the deployment of the WALD inbound extension manager | 149 |
| Globalization | 150 |
| Support for globalization | 150 |
| Adapter messages. | 151 |
| Related information | 151 |
| Video samples | 152 |

Notices 155

| | |
|---|-----|
| Programming interface information | 157 |
| Trademarks and service marks | 157 |

Index 159

Chapter 1. Overview of WebSphere Adapter for Lotus Domino

With WebSphere® Adapter for Lotus® Domino®, you can create modules that can exchange information with the Lotus Domino server. The adapter provides bidirectional exchange of business data between Service Component Architecture (SCA) applications and the Lotus Domino server. The business data can be Domino documents such as a calendar entry, to-do item, or a note.

The adapter enables two-way communication between an application running on IBM® Business Process Manager or WebSphere Enterprise Service Bus and a database. Using the adapter, an application can send requests to create, update, delete, or retrieve the documents in a database. The adapter has a built-in mechanism to support Enterprise Metadata Discovery (EMD), complex and rich text fields, and attachments in a document.

The adapter is imported and configured in a module that is created in your development environment and deployed to your application server or integration broker. After configuration, the adapter acts like a service provider in part of a service-oriented architecture (SOA) implementation and provides operations to retrieve or create documents in a Domino database. Client applications interact with the module instead of directly with the Lotus Domino server. This configuration gives the benefit of shielding the authentication details (such as user name and password) that you provide when you set up a module from the 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 performed.

With WebSphere Adapter for Lotus Domino software, you can automatically generate an interface to the Lotus Domino server to hide the lower-level details of the function. Depending on how you want to use the adapter, you can embed it with the deployed module, or install the adapter as a stand-alone component, to be used by more than one application. The adapter is deployed to IBM Business Process Manager or WebSphere Enterprise Service Bus. The application component interacts with the adapter instead of with the Lotus Domino server.

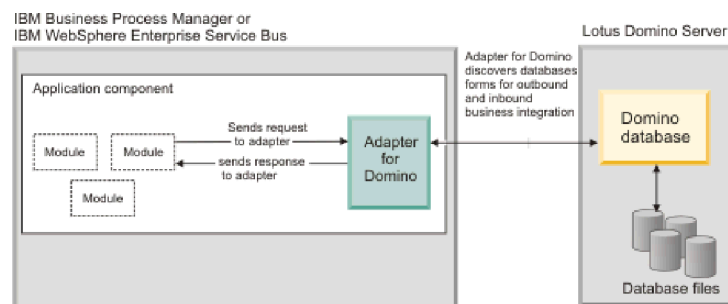


Figure 1. An application component calls the WebSphere Adapter for Lotus Domino, and the adapter interacts with the Lotus Domino server to obtain the data.

The adapter, which you generate with the external service wizard of Integration Designer, uses a standard interface and standard data objects. The adapter takes the standard data object sent by the application component and calls the Lotus Domino server. The adapter then returns a standard data object to the application

component. The application component does not have to deal directly with the Lotus Domino server. It is the WebSphere Adapter for Lotus Domino that calls the function and returns the results.

The module that you create 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 Lotus Domino server.

Hardware and software requirements

The hardware and software requirements for WebSphere Adapters are provided on the IBM Support website.

To view hardware and software requirements for WebSphere Adapters, see <http://www.ibm.com/support/docview.wss?uid=swg27006249>.

Additional information

The following links provide additional information you might need to configure and deploy your adapter:

- The compatibility matrix for WebSphere Business Integration Adapters and WebSphere Adapters identifies the supported versions of required software for your adapter. To view this document, go to the WebSphere Adapters support page: http://www-947.ibm.com/support/entry/portal/Overview/Software/WebSphere/WebSphere_Adapters_Family.
- Technotes for WebSphere Adapters provide workaround and additional information that are not included in the product documentation. To view the technotes for your adapter, go to the following Web page, select the name of your adapter from the **Product category** list, and click the search icon: <http://www.ibm.com/support/search.wss?tc=SSMKUK&rs=695&rank=8&dc=DB520+D800+D900+DA900+DA800+DB560&dtm>.

Technical overview of WebSphere Adapter for Lotus Domino

The adapter supports bidirectional exchange of business data between business applications and the Lotus Domino server. The adapter provides outbound and inbound processing under the Java 2 Platform, Enterprise Edition (J2EE) Connector Architecture (JCA) and integrates with Service Component Architecture (SCA) components.

The adapter provides the means for services running on IBM Integration Designer to exchange documents with one or more Domino servers. The services are contained in a module, which consists of both a project in IBM Integration Designer and a unit of deployment to IBM Business Process Manager. The module is packaged and deployed to IBM Business Process Manager as an enterprise archive (EAR) file.

Overview of the Domino products

The important constituents of the Lotus Domino suite of products are:

- Lotus Domino server: The Lotus Domino server provides a document database, a collaboration application server, and an e-mail server.
- Lotus Domino Administrator client: It is the administration tool of the Lotus Domino server.

- Lotus Domino Designer client: It is used for designing Lotus Domino applications.

Lotus Domino server metadata

WebSphere Adapter for Lotus Domino interacts with the Lotus Domino server using the metadata constituents.

- Database: The Lotus Domino server provides the built-in “NSF” format database, which contains all documents, forms, and scripts.
- Document: A document contains data in its various fields. There can be several types of documents in a database.
- Form: A form defines the structure of a document; the types of fields that a document contains. The business object structure in the adapter corresponds to a form in the Domino database.
- Item: The field in a document is called an item. The item has an item name, an item type, and one or multiple item values associated with it. A field in the business object is mapped to an item in the document. When the item is allowed to have multiple values, the mapped field in the business object is an array.

Outbound processing

WebSphere Adapter for Lotus Domino supports outbound request processing. When the adapter receives a request, which is sent in the form of a business object from the module, it processes the request and returns the result as a business object. The adapter provides a standard set of outbound operations.

Outbound processing begins when a module receives a business object from a service, which is called a request. When a request is received, the adapter uses the information stored in the business object and metadata from the interaction specification properties to perform an operation on the Lotus Domino server.

The adapter supports the following outbound operations:

- Create
- Update
- Delete
- Exists
- Retrieve
- RetrieveAll

Figure 2 on page 4 shows how the adapter and module function together as part of an outbound service. A module created for outbound processing receives the request. The adapter determines the operation to be performed based on the information received in the metadata from the interaction specification properties. It then performs the operation on the Lotus Domino server.

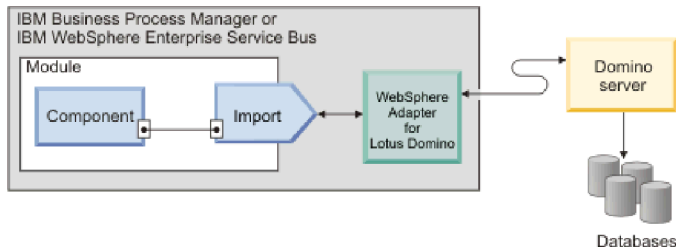


Figure 2. WebSphere Adapter for Lotus Domino as part of an outbound SOA implementation

The adapter connects to the Lotus Domino server with the server address, Domino Internet Inter-ORB Protocol (DIIOP) port number, user name, and password information. After connecting to the Lotus Domino server, the adapter lists all databases in the connected server. In each database, there can be several types of documents. Each type of document is called a form. The adapter lists all forms in a database and allows user to select the forms as business object types. For each selected business object type, method binding artifacts are generated to support outbound operations.

Note: All business data and metadata are stored in the Domino databases.

Outbound operations

Application components use operations to perform actions, such as retrieving data from databases in the Lotus Domino server. The adapter provides certain outbound operations. The name of the operation typically indicates the type of action that the adapter takes, such as Create or Update.

During outbound processing, the calling component (part of SCA application) calls the adapter to connect and work with business data on the Lotus Domino server. All interactions between the calling component and the server are request-response type.

Related concepts

“Business object structures” on page 17

A business object is a structure that consists of data and additional instructions, if any, for processing the data. The adapter supports the form-based business object structure. A form-based type is a content-specific business object that supports specific business object structures (such as, create and retrieve business objects).

Create operation:

The Create operation creates a document in the Lotus Domino server with the specified content.

The adapter receives a request in the form of a business object to create a document in the Lotus Domino server. The adapter processes the request and a document with a Universal ID is created in the specified database of the Lotus Domino server. The content of the document comes from the attribute values of the request business object.

After the document is created, the document content is sent back to the calling component as a response to indicate that the document was created successfully.

The response sent back to the calling component has the same business object structure as the request business object.

Note: To support attachments for rich text and rich text lite fields when creating documents, specify the `FilePath` of the attachments. It enables the adapter to get the source files by following the specified path.

When a processing error occurs, the Create operation might throw the `InvalidParentDocumentFault` fault. For more information about these business faults, refer to “Fault business objects” on page 114.

Related reference

“Fault business objects” on page 114

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.

Update operation:

The Update operation updates a document in the Lotus Domino server based on the request from the SCA.

The adapter receives the update request from the SCA application along with the Universal ID of the document. The adapter processes the request and makes the specified updates to the document.

Note: A document in the Lotus Domino server has a Universal ID and a Note ID associated with it. The adapter uses the value set in the Universal ID to search the document in the Domino database. If the Universal ID is not provided in the request business object, the adapter tries to use the value set in the Note ID. The adapter throws an error when both the Universal ID and the Note ID are not provided in the request business object.

By default, the adapter updates only those items in the document that have valid data in the business object. The remaining items are not updated in the document. The items in a document in the Lotus Domino server correspond to the fields in a business object of the WebSphere Adapter for Lotus Domino. During update, if the value of a business object field is explicitly set to null, the adapter updates the document by removing the item from the document.

Note:

- If a field in the business object is of `array([])` type, you can set the first value in the array as null to remove this item from the document in the Lotus Domino server.
- When updating the rich text or rich text lite fields, specify the `Overwrite` attribute to append or overwrite the fields.

After the document is updated, it is sent back to the calling component to indicate that the document was updated successfully.

The response sent back to the calling component has the same business object structure as the request business object. The response also has attributes that include the Universal ID and information about the result of the operation.

Note: No response documents (children) to the actual documents are removed during the Update operation.

The Update operation might throw the InvalidRequestFault, the DocumentNotFoundFault, or the InvalidParentDocumentFault fault depending upon the processing error. For more information about these business faults, refer to “Fault business objects” on page 114.

Related reference

“Fault business objects” on page 114

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.

Exists operation:

The Exists operation checks if a document exists in the Lotus Domino server.

The adapter receives the request to check if a document is available in the database along with the Universal ID of the document. The adapter processes the request and checks if the document is present in the Lotus Domino server.

Note: A document in the Lotus Domino server has a Universal ID and a Note ID associated with it. The adapter uses the value set in the Universal ID to search the document in the Domino database. If the Universal ID is not provided in the request business object, the adapter tries to use the value set in the Note ID. The adapter throws an error when both the Universal ID and the Note ID are not provided in the request business object.

The response business object is sent back to the calling component. The attributes in the business object include the Universal ID and information about the result of the operation.

When there is a processing error, the Exists operation might throw the InvalidRequestFault fault. For more information about these business faults, refer to “Fault business objects” on page 114.

Related reference

“Fault business objects” on page 114

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.

RetrieveAll operation:

The RetrieveAll operation is used to retrieve the documents from the database in the Lotus Domino server that matches the values specified in the search fields.

You can specify one or more of the following criteria in the business object sent with the outbound request.

- **Document business data:** When you specify document business data in the business object, the adapter reads all the valid data from the business object. The data are represented as items in the Domino document. The adapter searches the database for documents that match these valid data. All the matching documents that can be represented using the business object are returned back as response to the calling component.
- **ParentDocumentUNID:** When you specify a valid ParentDocumentUNID in the business object, the adapter retrieves all the child documents to this parent

document. All the matching documents which can be represented using the business object are returned back as response to the calling component.

- **No data set:** When you do not provide a data set in the business object, the adapter retrieves all the documents that can be represented using the business object. The adapter matches the fields in the documents with the stored values in the business object.

The retrieved documents are populated in a container business object and sent back to the calling component. The container business object has the following structure:

```
<DatabaseName><FormName>Container --> <DatabaseName><FormName> []
```

Search implementation using database search method

The RetrieveAll operation supports searching of documents based on the following types of search criteria that you can specify during the run time:

- **Single values:** You can specify a single value in the search criteria. For example, the adapter can retrieve documents based on the single word, number, or date specified in the search field. You can find more information about the search values that can be specified in the Number and Date type fields in the following points.
 - **Numbers:** You can specify a double digit number or an integer value as the search criteria in the Numbers type field. You can use this search function to retrieve documents where a Number type field equals to a certain value. For example, to retrieve documents where employee salary equals to "1234.56", enter the value "1234.56" in the EmployeeSalary field.
 - **Date:** You can specify a date as the search criteria in any Date type field. For example, to retrieve documents where birthday equals to "1982/11/22", enter the value "1982/11/22" in the Birthday field.

Note: The Date format depends on the client that starts the WebSphere Adapter for Lotus Domino. The client can be JavaServer Pages (JSP), a Java program, or a IBM Integration Designer test component.

- **Multiple values:** You can specify multiple values as the search criteria for any Lotus Domino field types. This feature can be used to find the documents with the specified values at run time. For example, you can configure the adapter to retrieve documents that match any one of the words from the multiple words specified in the search criteria.
- **Text search for rich text contents:** You can specify the text in order to search documents that have rich text contents at run time. The adapter conducts a search by using the text specified in the RichText field of a document during the run time. During the RetrieveAll operation, all matching documents are retrieved by the adapter. The adapter supports both single and multiple text content searches for the RichText field. You can also provide wildcard suffixes, such as "*" for searching documents in the RichText field.
- **Search using multiple fields:** You can specify search values in more than one field. In this configuration, the adapter uses the logical AND operator to group the search strings. For instance, you can specify a single value, multiple values, RichText, or ParentDocumentUNID in different fields. The adapter returns only that search content that matches all the specified criteria.

Search implementation using indexed search method

Indexed search: You can specify index-based search to enable faster text searching, if the target Domino database is full-text indexed. The adapter uses the FTSearch function provided by the Lotus Domino client API. The FTSearch method searches for documents using the full text-index in a database that matches a queried text. The adapter returns exact matches of the search string followed by closer matches.

Note: The indexed search method does not work for RichText, Attachments, and Date fields.

The indexed search method also supports search using multiple fields. You can specify search values in more than one field. In this configuration, the adapter uses the logical AND operator to group the search strings. For instance, you can specify a single value, multiple values, or ParentDocumentUNID in different fields. The adapter returns a logical AND operation of all the specified search criteria, returning exact and close matches for string type fields and exact matches for numeric type fields.

The following are the differences between indexed search and database search methods:

- When the database is fully indexed, the indexed search method is faster compared to the database search method.
- The accuracy of the results returned by the indexed search depends on the updated database indexes.
- The database search returns only the exact matches; whereas indexed search returns exact matches first up in the searched list and then the close matches.

Faults occurrence during the RetrieveAll operation

When there is a processing error, the RetrieveAll operation might throw the InvalidParentDocumentFault or the MatchesExceededLimitFault fault. The InvalidParentDocumentFault fault can occur when an invalid ParentDocumentUNID is specified in the interaction specification. The MatchesExceededLimitFault fault can occur when the matching documents exceed the specified MaxRecords value. For more information about these business faults, see “Fault business objects” on page 114.

Related reference

“Fault business objects” on page 114

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.

“Interaction specification properties” on page 127

Interaction specification properties control the interaction for an operation. The external service wizard sets the interaction specification properties when you configure the adapter. Typically, you do not need to change these properties. However, you can change some properties for outbound operations. To change these properties after the application is deployed, use the assembly editor in IBM Integration Designer. The properties are located in the method binding of the import.

Retrieve operation:

The Retrieve operation extracts a document content based on the request from the SCA.

The adapter receives the retrieve request along with the Universal ID of the document. The adapter processes the request and extracts the content of the document from the database in the Lotus Domino server.

Note: A document in the Lotus Domino server has a Universal ID and a Note ID associated with it. The adapter uses the value set in the Universal ID to search the document in the Domino database. If the Universal ID is not provided in the request business object, the adapter tries to use the value set in the Note ID. The adapter throws an error when both the Universal ID and the Note ID are not provided in the request business object.

After the content of the document is retrieved, it is sent back as the response to the calling component. The response has the same business object structure as the request business object. The attributes in the business object include the Universal ID and information about the result of the operation.

When there is a processing error, the Retrieve operation might throw the `InvalidRequestFault` or the `DocumentNotFoundFault` fault. For more information about these business faults, refer to “Fault business objects” on page 114.

Related reference

“Fault business objects” on page 114

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.

Delete operation:

The Delete operation deletes a document in the Lotus Domino server based on the request from the SCA.

The adapter receives the delete request along with the Universal ID of the document. The adapter processes the request and removes the document from the Domino database.

Note: A document in the Lotus Domino server has a Universal ID and a Note ID associated with it. The adapter uses the value set in the Universal ID to search the document in the Domino database. If the Universal ID is not provided in the request business object, the adapter tries to use the value set in the Note ID. The adapter throws an error when both the Universal ID and the Note ID are not provided in the request business object.

The response business object is sent back to the calling component. The attributes in the business object include Universal ID and information about the result of the operation.

Note: If there are any response documents (children) to the actual document, then all the child documents are also removed from the Lotus Domino server.

When there is a processing error, the Delete operation might throw the `InvalidRequestFault` or the `DocumentNotFoundFault` fault. For more information about these business faults, refer to “Fault business objects” on page 114.

Related reference

“Fault business objects” on page 114

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.

Inbound processing

WebSphere Adapter for Lotus Domino supports inbound processing of events. The adapter polls the event database in the Lotus Domino server for events at specified intervals. Each time a form is created or updated, the adapter tracks the process as an event. When the adapter detects an event, it requests a copy of the form, converts the form data into a business object, and sends it to the consuming service.

During inbound processing, the adapter first establishes a connection with the Lotus Domino server. The adapter then polls the event database in the Lotus Domino server at regular intervals for new events. When a document in a database in the Lotus Domino server is created, updated or deleted, the event monitoring agent of the adapter monitors the event and adds that event in the event database. The events related to a document in the database are captured in the same form that also contains the document. The form contains the business object based on the type of an event.

The following illustration shows the inbound processing flow for WebSphere Adapter for Lotus Domino. The adapter polls the Lotus Domino server for incoming events, converts them into business objects, and then the adapter sends them to a consuming service.

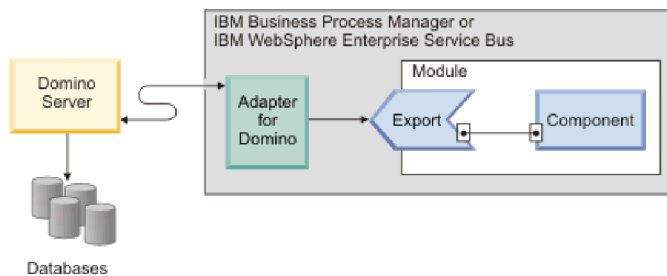


Figure 3. Inbound processing

Note: The adapter provides an extension manager to monitor the create, update, or delete events of a document in the database in the Lotus Domino server. You can configure the extension manager with a document in the Domino database.

When you create, update, or delete a document, the extension manager inserts an event record into the event table to record that event. When the adapter is polling events, it reads the information from this event record and retrieves the relative document. The adapter then returns the document as a business object. For an event type of either create or update, the adapter returns the document as a business object on the adapter workstation. For a delete event, the returned business object contains only the Universal ID of the deleted document.

Note:

- During inbound processing, an exception in the event table EventTab.nsf is handled by the configured exception handler.
- If you want to filter only those events that you have subscribed to, specify the EventTypeFilter property of the activation specification properties. If the EventTypeFilter property is set, the adapter delivers only those events that are in the filter list.

The adapter then forwards the business object to the endpoint which is an exposed interface from an SCA module that offers a business service to the outside world. Business objects contain the information that the adapter obtains from a document in a form structure.

Use the following steps to process the inbound events:

1. Establish a connection to the Lotus Domino server.
2. Poll events from the event table.
3. Retrieve the relative business object based on the event type. For a delete event type, only the Universal ID of the form is retrieved.
4. Deliver the business object to the endpoint.

Support for event based filtering

The adapter supports filtering of events based on the operations you select during the inbound processing. You can select the events to be processed for a single business object in the event table during the Enterprise Metadata Discovery (EMD) phase.

During run time, only the events corresponding to the specified operations are polled by the adapter. For instance, the filter can be specified for Create, Update, and Delete events or only the Create events of a business object, for processing during the inbound operation.

The filtering of events by the adapter ensures that only those events are polled for which the method bindings are created during the EMD phase. This results in successful delivery of events by the adapter.

Support for RichText and Attachments

During run time, the WebSphere Adapter for Lotus Domino supports retrieval of rich text and attachments. For an event type of either create or update, the business objects containing a RichText field retrieves texts and attachments. If the business objects contain the Attachments field, the value in the hexBinary form is retrieved.

Related tasks

“Setting deployment properties for a runtime environment” on page 63

The connection properties are initialized to the values that the wizard used to connect to the database. To configure the module to use other values, change values such as user name and password here.

“Setting location properties for artifacts” on page 69

After you select and configure deployment properties, use the external service wizard to configure location properties that the adapter uses to generate the service.

Related reference

“Activation specification properties” on page 135

Activation specification properties are properties that hold the inbound event processing configuration information for a message endpoint.

“Resource adapter properties” on page 119

The resource adapter properties control the general operation of the adapter, such as specifying the namespace for business objects. You set the resource adapter properties using the external service wizard when you configure the adapter. After deploying the adapter, use the administrative console to change these properties.

“Inbound configuration properties” on page 130

WebSphere Adapter for Lotus Domino 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 IBM Integration Designer or the administrative console, but connection properties for the external service wizard cannot be changed after deployment.

Event store

The event store is a persistent cache where event records are saved until the polling adapter can process them. To keep a track of inbound events as they make their way through the system, the adapter uses an event store.

Each time you create, update, or delete an event record in the Lotus Domino server, it is referred to as an event. The adapter updates the status of the event in an event store, when ever a business object is created, updated, or deleted. For recovery purposes, the status of each event is continually updated by the adapter until the events are delivered to a configured endpoint in the IBM Business Process Manager or WebSphere Enterprise Service Bus.

For example, an update to the document in the Lotus Domino server triggers the adapter to record the event in the event table. The event record stores information associated with the event such as the event status. The adapter can show the status of an event as 0 for unhandled events, 1 for processed events, or -1 for the failed events. The adapter retrieves this event and the update to the document. The event store implementation by the adapter ensures that data is correctly retrieved from the Lotus Domino server. The adapter then processes the event and delivers the event to the configured endpoint.

Each event store created by the adapter is associated with a particular module; however the same event store can be used by multiple modules. If you want multiple modules to use the same event store, you must set the `EventTypeFilter` property to a different event type for each module using the event store. You can configure the `EventTypeFilter` property in the `ActivationSpec` instance for each module.

During inbound processing, the adapter polls the event business components from the event store at regular intervals. Each time it polls, a number of events are processed by the adapter. Events are processed in ascending order of priority and ascending order of the event time stamp. In each poll cycle, new events are picked up. The adapter retrieves the value set in the EventID field of the event table and loads the business object that corresponds to it. The business object is created from the retrieved information and is delivered to the endpoints.

If you set the activation specification property AssuredOnceDelivery to true, a transaction ID (XID) value is set for each event in the event store. After the event is retrieved for processing, the XID value for the event is updated in the event store. The updated value is displayed in the XID column in the event business component. The event is then delivered to its corresponding endpoint, and the status is updated to show that the event has been successfully delivered. If the application is stopped or the event is not processed completely, the XID column is filled with a value. This process ensures that the event is reprocessed and sent to the endpoint. After the connection is reestablished or the adapter starts again, the adapter checks for events in the event store that have a value in the XID column. The adapter processes these events first and then polls the other events during the poll cycles.

The adapter can either process all the events or process events filtered by business object type. You set the filter through the activation specification property, EventTypeFilter. The property contains a comma-delimited list of business object types. Only the types specified in the property are processed. If the EventTypeFilter property is not set, then all events are processed. If the FilterFutureEvents property is set to true, the adapter filters events based on the time stamp. The adapter compares the system time in each poll cycle to the time stamp on each event. If an event is set to occur in the future, it is not processed until that time.

After an event is successfully posted and delivered to the endpoint, the entry is deleted from the event store. Failed events (posting and delivery to the endpoint is unsuccessful), remain in the event store and are marked -1. This feature prevents duplicate processing of the events.

Event store structure for Domino business objects and business components

The event business component stores information about the event. The information stored is used by the resource adapter during event subscription to build the corresponding business object and send it to the registered endpoints. The information that is stored, as well as the structure of the event store used by the adapter, is shown in the following table.

Table 1. Event store structure

| Field | Description | Example |
|---------------|---|--|
| Pick Fields | The fields which the user is interested in. This field is not configurable. | Note ID |
| Event ID | The ID of the event row. | Automatically generated unique ID in Domino (for example: 1-XYZ) |
| Event type | The type of event. | Create, Update, or Delete |
| Database path | The database path of the document. | test/doc.nsf |

Table 1. Event store structure (continued)

| Field | Description | Example |
|-----------------|--|---------------------|
| Form | The Form is an instance of the document on which the event occurs. | Response |
| UNID | Universal ID of the document | |
| Status | <p>The event status that is initially set to the value for a new event and updated by the adapter as it processes the event. The status can have one of the following values:</p> <ul style="list-style-type: none"> • 0: Identifies a new event. • 1: Identifies an event that has been delivered to an endpoint. • -1: An error occurred while processing the event. <p>This column cannot be null.</p> | 0 |
| Event timestamp | The time stamp for the event. The format is in <i>mm/dd/yyyy hh:mm:ss</i> . | 02/24/2007 11:37:56 |
| Connector ID | The ID set for the Domino adapter to filter events. This field is mapped to the AdapterInstanceFilter property of the activation specification properties. | Domino001 |
| XID | The transaction ID. It ensures the <i>assured once delivery</i> of the events. | None |

Event recovery

The adapter supports event recovery for inbound processing in case of abrupt termination. During inbound processing, you can configure the recovery mechanism based on your requirements.

To use the recovery feature provided by IBM Business Process Manager, you must set the AssuredOnceDelivery property in the activation specification to true. If it is set to false, failed events cannot be recovered. Duplicate events can be delivered if AssuredOnceDelivery is set to false. To improve performance, you can set the AssuredOnceDelivery properties to false.

You implement the event recovery mechanism provided by the Domino adapter by setting the failedEventRetryLimit property in the activation specification. When the Lotus Domino server is down, the adapter tries to recover the event based on the number of times specified in the failedEventRetryLimit property.

Inbound extension manager

During inbound processing, WebSphere Adapter for Lotus Domino (WALD) inbound extension manager monitors the operations on the Lotus Domino server. Whenever you create, update, or delete a document from the Domino database, the inbound extension manager generates corresponding events containing the operation information.

The WALD inbound extension manager is a dynamic linked library that monitors the create, update, and delete events on the Lotus Domino server. The extension manager is deployed on the Lotus Domino server. You configure the extension manager according to the operating system on which the Lotus Domino server is

run. You require the waldinbound.nsf (event table and configuration table database) and the operating system-specific library files in the Domino database to configure the extension manager.

The WALD Inbound Event Table in the waldinbound.nsf database contains the configuration parameters for the inbound extension manager. The following views are available for configuring the inbound event table.

- **ConfigurationView:** The view was used to configure the inbound event table. This view is now deprecated.
- **EventTriggerConfigurationView:** This view is used to create or edit event trigger rules for a monitored Form.
- **SystemConfigurationView:** This view is used to edit the existing logging configuration of the inbound event table.

The adapter supports configuring of multiple documents with either the same or with different Connector IDs in the EventTriggerConfigurationView view.

Same Connector ID

The multiple documents with a same Connector ID are merged to create a single trigger rule. The following examples show the configuration for using a same Connector ID in two documents.

- Configuration document 1: Connector1; Create, Update; DB1a, DB1b
- Configuration document 2: Connector1; Create, Update; DB1a, DB1c

Here, Connector1 is the Connector ID, Create and Update are the event types, and DB1a, DB1b, and DB1c are the databases.

The following sample scenarios show the event logging behavior for a same Connector ID.

- When a document is created or updated in DB1a database, one event is logged for Connector1.
- When either a document is created in DB1b or DB1c database, only one event is logged for Connector1 because both the databases have the same Connector ID.

Different Connector IDs

The adapter supports multiple configuration documents in the Event Trigger Configuration view of Extension Manager. The following examples show the configuration for using different Connector IDs in two documents.

- Configuration document 1: Connector1; Create, Update; DB1a, DB1b
- Configuration document 2: Connector2; Create, Update; DB1a, DB1c

Here, Connector1 and Connector2 are the Connector IDs, Create and Update are the event types, and DB1a, DB1b, and DB1c are the databases.

The following sample scenarios show the event logging behavior for different Connector IDs.

- When a document is created or updated in DB1b database, one event is logged for Connector1.
- When a document is created or updated in DB1a database, two events are logged; one for Connector1 and the other for Connector2.
- When a document is created or updated in DB1c database, one event is logged for Connector2.

The dynamic linked library file monitors the create, update, and delete events in the event directory. The following table lists the dynamic linked library file required for the different operating systems.

Table 2. Operating system-specific dynamic linked library file

| Dynamic linked library file name | Operating system |
|----------------------------------|------------------|
| waldinbound.dll | Windows |
| libwaldinbound_r.a | AIX® |
| libwaldinbound.so | Linux |
| libwaldinbound.so | Solaris |

Operating systems supported by WALD inbound extension manager

The extension manager libraries required to support 32-bit Lotus Domino server on specific operating systems are shown in Table 2.

Table 3. WALD inbound extension manager libraries for 32-bit Domino on supported operating systems.

| Lotus Domino server | Library directory after installation | Library file name |
|--|--------------------------------------|--------------------|
| 32-bit Domino on Microsoft Windows | dependencies/win32 | waldinbound.dll |
| 32-bit Domino on IBM AIX | dependencies/aix32 | libwaldinbound_r.a |
| 32-bit Domino on Novell SUSE Linux Enterprise Server | dependencies/suse32 | libwaldinbound.so |
| 32-bit Domino on Red Hat Enterprise Linux | dependencies/redhat32 | libwaldinbound.so |
| 32-bit Domino on Solaris Operating System | dependencies/solaris32 | libwaldinbound.so |

Note: The current version of WALD inbound extension manager does not support the 64-bit Lotus Domino server.

For detailed information about the deployment and configuration information of the WALD inbound extension manager, see the topic “Deployment and configuration of the WALD inbound extension manager” on page 144. After you configure the inbound extension manager, see the topic “View and verify the deployment of the WALD inbound extension manager” on page 149 to verify the extension manager configuration.

Related reference

“Deployment and configuration of the WALD inbound extension manager” on page 144

The adapter provides for deployment and configuration of the WALD inbound extension manager on the Lotus Domino server. When you deploy the extension manager, the adapter is able to monitor the create, update, and delete events in the Domino database.

“View and verify the deployment of the WALD inbound extension manager” on page 149

After you have deployed the inbound extension manager, you can view and verify its deployment to check if the extension manager is correctly configured to monitor the events in the Lotus Domino server.

Business object structures

A business object is a structure that consists of data and additional instructions, if any, for processing the data. The adapter supports the form-based business object structure. A form-based type is a content-specific business object that supports specific business object structures (such as, create and retrieve business objects).

Depending on the custom form types and operations when you create an external service, the adapter generates the corresponding business object definition. For example, if you select RetrieveAll operation, the adapter generates a container business object. The container business object contains the matching business objects. The following graphic illustrates this relationship:

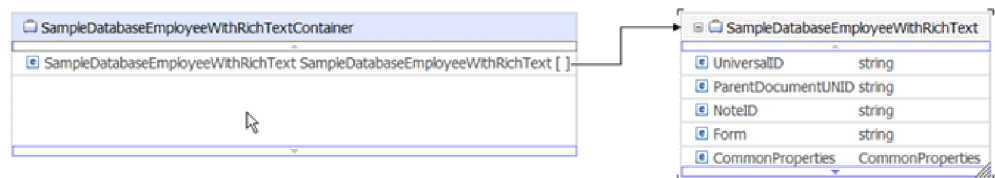


Figure 4. RetrieveAll container business object

Form-based business object structure

The form-based business object structure consists of attributes that directly relate to the fields in a document. By selecting the form document in the external service wizard, you can use the Integration Designer to generate form-based business object structure at run time. Optionally, if you choose, it can also contain child business objects for a container business object and attachments. The following illustration shows an example of a form-based business object generated from the form document in the sample Domino database.

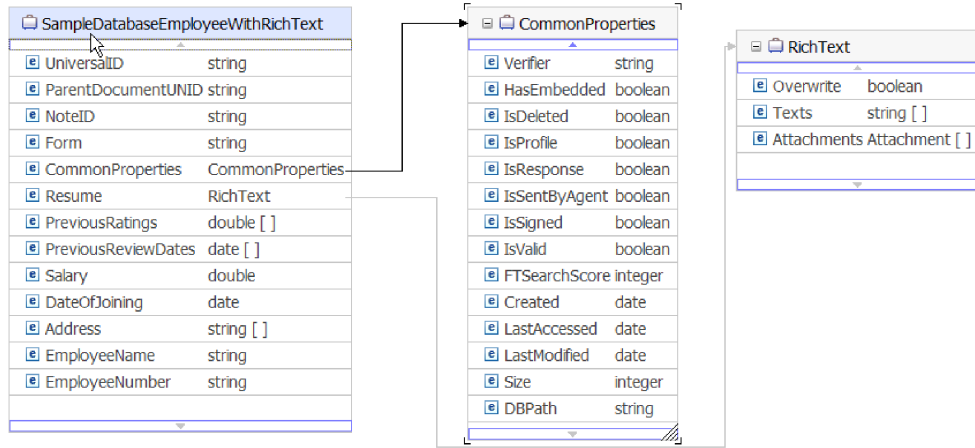


Figure 5. Retrieve business object

The following attributes are common for all parent business objects:

- Universal ID
- ParentDocumentUNID
- Note ID
- CommonProperties

Note: If a form contains fields with attributes that have identical names as the standard attributes present in the top-level business object, then when the business object is generated for this form, the attributes of this form field is named as FormItem<FormFieldName>. For example, for CommonProperties business object, the name changes to FormItemCommonProperties.

In Figure 5, CommonProperties and RichText are the two child business objects. The CommonProperties business object stores the metadata information for the parent business object. The RichText business object contains information for managing attachments. The attachment data is stored in an Attachment business object.

RichText business object structure

The adapter generates RichText business objects for both the Domino Rich Text or Rich Text Lite item types. For information about the rich text attributes see, “Business object attributes for Rich Text data type” on page 112. The following illustration shows the structure of a RichText business object.

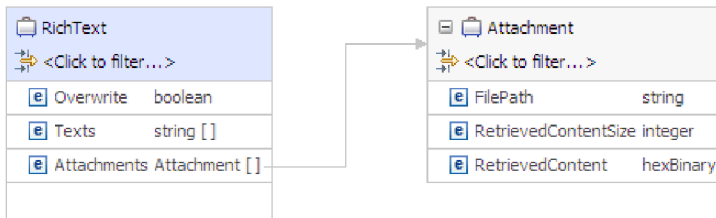


Figure 6. RichText business object

RichTextComplexItem business object structure

For complex objects, the RichTextComplexItem business object is generated as a child business object of the RichText or RichText Lite business objects. For information about the rich text complex item attributes see, “Business object attributes for Rich Text complex item data type” on page 112. The following illustration shows the structure of a RichTextComplexItem business object.

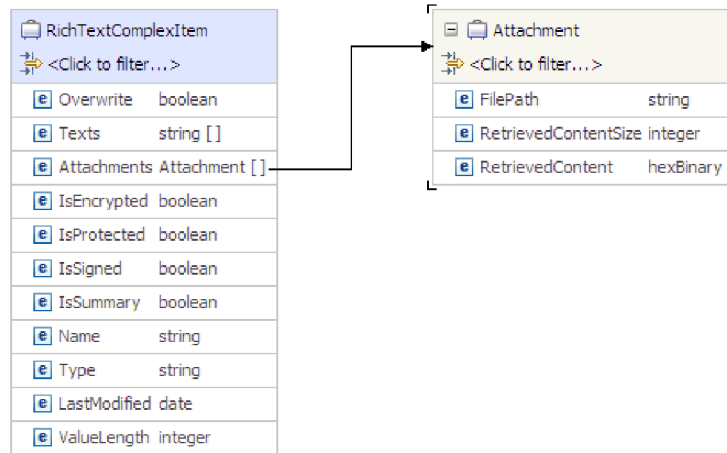


Figure 7. RichTextComplexItem business object

DeleteResponse business object structure

During the Delete operation, the status of the operation is checked using the DocumentDeleted attribute in the DeleteResponse business object. The following illustration shows the structure of the DeleteResponse business object.

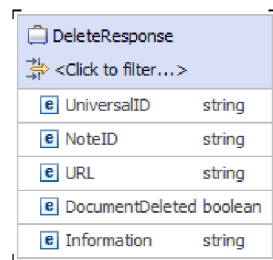


Figure 8. DeleteResponse business object

ExistsResponse business object structure

During the Exists operation, the status of the operation is checked using the DocumentExists attribute in the ExistsResponse business object. The following illustration shows the structure of the ExistsResponse business object.

| ExitsResponse | |
|----------------------|---------|
| <Click to filter...> | |
| UniversalID | string |
| NoteID | string |
| URL | string |
| DocumentExists | boolean |
| Information | string |

Figure 9. ExitsResponse business object

Related reference

“Outbound operations” on page 4

Application components use operations to perform actions, such as retrieving data from databases in the Lotus Domino server. The adapter provides certain outbound operations. The name of the operation typically indicates the type of action that the adapter takes, such as Create or Update.

“Business object information” on page 111

A business object is a structure that contains the data required by the adapter to perform an operation. The name of the business object is generated by the wizard in accordance with the naming convention for the adapter.

“Interaction specification properties” on page 127

Interaction specification properties control the interaction for an operation. The external service wizard sets the interaction specification properties when you configure the adapter. Typically, you do not need to change these properties. However, you can change some properties for outbound operations. To change these properties after the application is deployed, use the assembly editor in IBM Integration Designer. The properties are located in the method binding of the import.

External software dependencies

The adapter needs to have some dependency files to communicate with the Lotus Domino server.

The required Domino libraries and dependency files needed to connect to the Lotus Domino server are bundled as a part of the adapter.

The external service wizard

The external service wizard in WebSphere Adapter for Lotus Domino is used to create services and to generate business objects from the selected objects. The wizard also generates the module and the service artifacts that enable the adapter to run as a Service Component Architecture (SCA) component.

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 DMNORA plus the value of the adapter ID property. For example, if the adapter ID property is set to 001, the component ID is DMNORA001.

If you run multiple instances of the same adapter, ensure that the first characters of the adapter ID property are unique for each instance so that you can correlate the 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 the log and trace information to a particular instance of an adapter. For example, when you set the adapter ID property of two instances of WebSphere Adapter for Lotus Domino to 001 and 002. The component IDs for those instances, DMNORA001 and DMNORA002, 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. If you set the adapter ID properties of two instances to Instance01 and Instance02, you will not be able to examine the log and trace information for each adapter instance because the component ID for both instances is truncated to .

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. It prevents 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.

For more information, see the “Adapter ID” on page 119 property.

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 IBM Business Process Manager and WebSphere Enterprise Service Bus support other types of faults, the adapter generates only business faults, which are called *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.

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 10 on page 22.

| WBIFault | |
|---------------|-----------------------|
| message | string |
| errorCode | string |
| primaryKeySet | PrimaryKeyPairType [] |

Figure 10. The structure of the WBIFault business object

Some faults contain attributes such as the matchCount, NotesErrorID, NotesErrorText, and NotesErrorInternal to provide additional information about the error. For others, WBIFault contains all the information needed to handle the fault.

Related reference

“Fault business objects” on page 114

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.

WebSphere Application Server environment variables

WebSphere Application Server environment variables can be used in the external service wizard to specify directory values. You can now change any string property values in inbound and outbound configuration by only changing the environment variables.

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 IBM Business Process Manager or WebSphere Enterprise Service Bus administrative console.

In IBM Business Process Manager or WebSphere Enterprise Service Bus, 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 change the environment variable in the IBM Business Process Manager or WebSphere Enterprise Service Bus administrative console. For example, when setting the server IP address in the managed connection factory properties, you can use `${DOMINO_SERVER_IP}` as the variable name instead of hard coding it as `127.0.0.1`.

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 configurations.

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

- The name of the environment variable, for example, `DOMINO_SERVER_IP`.
- The value that the symbolic name represents, for example: `127.0.0.1`.

- 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. The Node is the default scope level.
 - Cell scope limits visibility to all servers on the named cell.

Related tasks

“Defining WebSphere Application Server environment variables” on page 41
Use the administrative console of IBM Business Process Manager or WebSphere Enterprise Service Bus to define WebSphere Application Server environment variables.

Related reference

“Managed (J2C) connection factory properties” on page 121
Managed connection factory properties are used by the adapter at run time to create an outbound connection instance with the Lotus Domino server.

“Activation specification properties” on page 135
Activation specification properties are properties that hold the inbound event processing configuration information for a message endpoint.

Chapter 2. Planning for adapter implementation

Before using WebSphere Adapter for Lotus Domino, make sure you understand the experience you need and the server environment in which it runs. Learn the considerations for deploying the adapter in your server environment, and find out how to improve the performance and availability of the adapter by using a clustered server environment.

Security

The adapter uses the J2C authentication data entry, or the authentication alias feature of Java 2 security to provide secure user name and password authentication. For more information about security features, see the documentation for IBM Business Process Manager or WebSphere Enterprise Service Bus.

Support for protecting sensitive user data in log and trace files

You can configure the adapter to prevent sensitive or confidential data, in the log and trace files, from being viewed by users without authorization.

Log and trace files for the adapter can contain data from your Lotus Domino server, which might contain sensitive or confidential information. Sometimes these files might 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 such situations, the adapter lets you specify whether 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 (J2C) connection factory properties” on page 121 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 UNID of the event record
- User name, Password, Host Name and DIOP port number

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

User authentication

The adapter supports several methods for supplying the user name and password that are needed to connect to the Lotus Domino 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.

Authentication at run time

At run time, the adapter needs to provide the user name and password to connect to the Lotus Domino 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. You can configure the adapter to get your user information, through any of the following methods:

- Adapter properties
- Connection specification properties
- 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 this 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 access that Lotus Domino server. This includes the adapters embedded in application EAR files as well as adapters that are separately installed on the server.

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 must 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 38](#)

An authentication alias is a feature that encrypts the password used by the adapter to access the Lotus Domino server. The adapter can use it to connect to the Lotus Domino server instead of using a user ID and password stored in an adapter property.

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 affect the type of deployment option you choose.

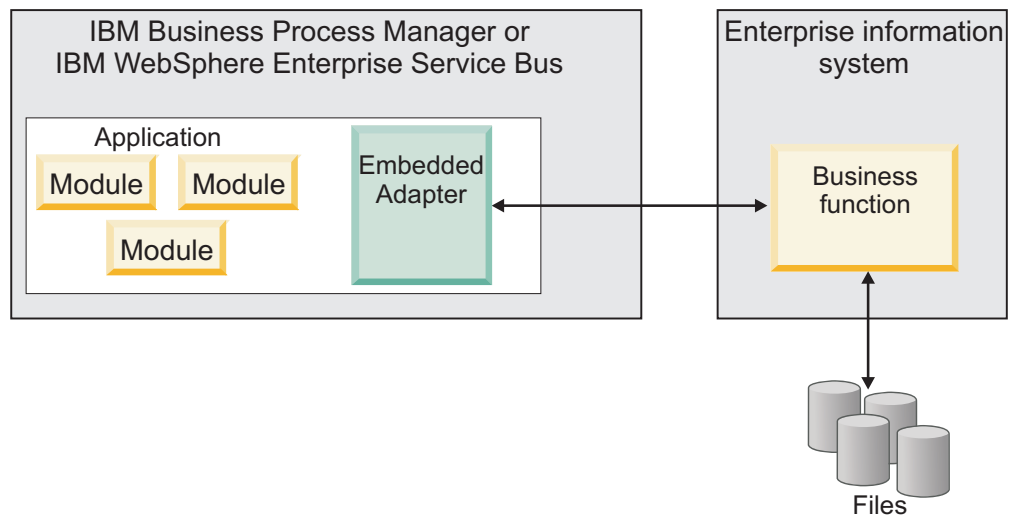
The following are the deployment options:

- **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

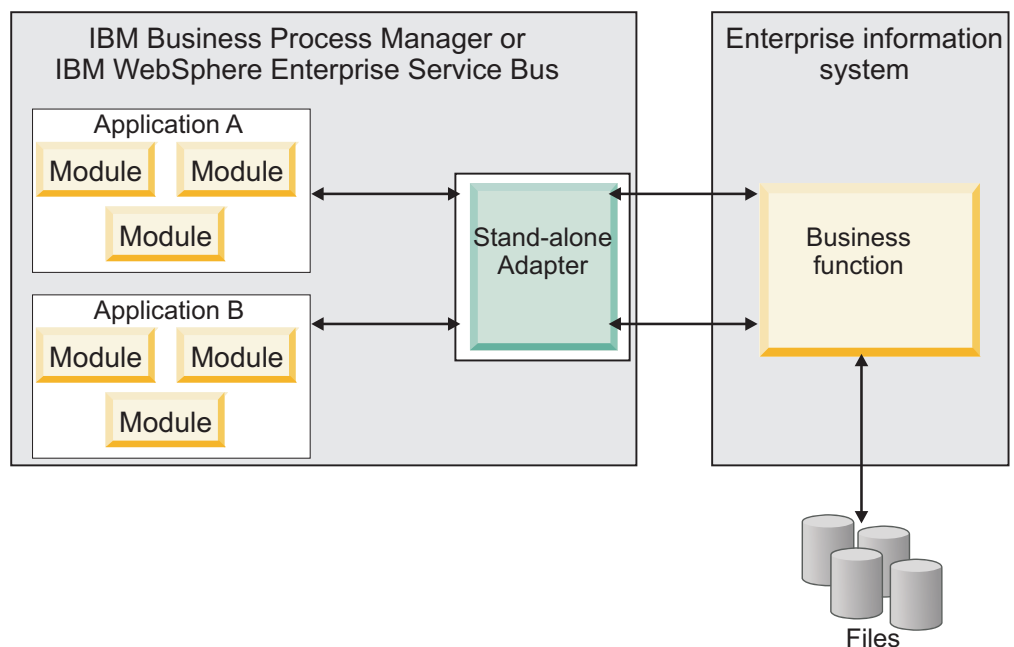
embedded adapter enables you to upgrade the adapter in a single module without the risk of destabilizing other modules by changing their adapter version.

- **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 IBM Integration Designer, you can choose how to package the adapter [either bundled with the (EAR) file or as a stand-alone (RAR) file]. Your choice affects how the adapter is used in the run time 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 to run 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

Consider the following items if you plan to embed 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 run time environments. *Class loader isolation* means that 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

Consider the following items if you plan to use 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.

Note:

When you install multiple adapters with different versions of CWYBS_AdapterFoundation.jar, and if a lower version of the CWYBS_AdapterFoundation.jar is loaded during runtime, the adapter will return the ResourceAdapterInternalException error message, due to a version conflict. For example, when you install Oracle E-Business Suite adapter version 7.0.0.3 and WebSphere Adapter for Lotus Domino version 7.5, the following error message is displayed: IBM WebSphere Adapter for Lotus Domino has loaded file:/C:/IBM/WebSphere/ProcServer7/profiles/ProcSrv01/installedConnectors/CWYOE_OracleEBS.rar/CWYBS_AdapterFoundation.jar with version 7.0.0.3. However, the base level of this jar required is version 7.5. When you install multiple adapters with different CWYBS_AdapterFoundation.jar versions, the adapter returns the ResourceAdapterInternalException message, due to a version conflict.

Considerations while deploying WebSphere Adapter 7.5 with another version

There are occasions when you have to work with embedded adapters that do not need a client-server communication, standalone adapters that need a server connection, or a hybrid mix of adapter connections.

The following scenarios cover the different behaviors of AFC version conflict detection.

Deploying a standalone Adapter

1. Install WebSphere Adapter for Flat Files version 7.0.1.0 through the IBM Business Process Manager administrative console.
2. Install WebSphere Adapter for SAP Software version 7.5.0.0 through the administrative console.
3. Create ActivationSpec for an ALE passthrough inbound operation.
4. Create an application in IBM Integration Designer for a standalone ALE passthrough inbound operation.
5. Install and start the application through the administrative console.
6. Verify the error.

Note: An error message will be generated in the log/trace area of IBM Business Process Manager, to indicate an AFC version conflict.

Deploying an embedded Adapter

1. Import a build of WebSphere Adapter for FTP version 7.0.1.0, using a RAR file.
2. Create a FTP Inbound EMD operation.
3. Import a build of WebSphere Adapter for Oracle E-Business Suite version 7.5.0.0, using a RAR file.
4. Create an Oracle inbound EMD operation, in the same module where you have created the FTP Inbound EMD operation.
5. Deploy the module to IBM Business Process Manager.
6. Check the trace.

At step 5, the deployment will fail. At step 6, you will get an internal error message due to the AFC version conflict.

Note: To avoid a name conflict between the business object generated by the two adapters, you may need to generate the artifacts into different folders.

Deploying a combination of standalone and embedded Adapters

1. Install WebSphere Adapter for JDBC version 7.0.1.0 through the IBM Business Process Manager administrative console.
2. Create an ActivationSpec for a JDBC inbound operation.
3. Create an application in IBM Integration Designer for a JDBC inbound operation, for the standalone Adapter deployment.
4. Deploy the JDBC inbound application and trigger your inbound events.
5. Create an application in IBM Integration Designer for a WebSphere Adapter for SAP Software version 7.5.0.0 inbound embedded Adapter deployment.
6. Deploy an SAP inbound application, and trigger your inbound events.

Note: You can resolve the AFC version conflict by using different class loaders for the standalone and embedded deployments. With this approach, the migration process will handle different CWYBS_AdapterFoundation.jar files and will not conflict with each other. You can start both JDBC and SAP inbound applications successfully, and process Inbound events without exception.

For further assistance, visit http://www-947.ibm.com/support/entry/portal/Overview/Software/WebSphere/WebSphere_Adapters_Family.

Migrating to version 7.5 of WebSphere Adapter for Lotus Domino

Migration considerations

WebSphere Adapter for Lotus Domino version 7.5 may have some features and updates that might affect your existing adapter applications. Before migrating applications that use WebSphere Adapter for Lotus Domino, you must consider some factors that might affect your existing applications.

Compatibility with earlier versions

WebSphere Adapter for Lotus Domino version 7.5 is fully compatible with the custom business objects (XSD files) and data bindings that are created using the adapter version 7.0 and enables the existing business objects and data bindings to work well in the latest version of the adapter.

The version 7.5 of WebSphere Adapter for Lotus Domino is fully compatible with version 7.0, and therefore the applications that used the version 7.0 runs unchanged when you upgrade to version 7.5. However, if you want your applications to use features and functionality present in version 7.5 of the adapter, run the migration wizard.

The migration wizard replaces (upgrades) version 7.0 of the adapter with version 7.5 and enables version 7.5 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 7.5 of the adapters. If any of your applications embed an adapter that is version 7.0 and you

are upgrading to version 7.5, and you want your applications to take advantage of the features and functions in version 7.5, you might need to change to those applications.

If the artifacts within a module have inconsistent versions, the entire module is marked as unavailable for migration and cannot be selected. Version inconsistencies are recorded in the workspace log, as they indicate that a project might be corrupted.

The adapter migration wizard in Integration Designer version 7.5 only supports the migration of adapters from version 7.0 to version 7.5.

Deciding whether to upgrade or to upgrade and migrate

If you decide that you want to upgrade the adapter from version 7.0 to version 7.5, 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 selecting any adapter artifacts installs and upgrades your adapter. As the artifacts are not migrated, your applications cannot take advantage of the features and capabilities that exist in version 7.5 of the adapter.

Migrating multiple adapters referred within a project

When you select the module project and launch the migration wizard:

- The **Source connector** field lists the connector projects with the selected module project.
- The **Dependent artifact projects** area lists only the selected module project.

If you select the connector project and launch the migration wizard:

- The **Source connector** field lists only the selected connector project.
- The **Dependent artifact projects** area lists all projects which reference the selected connector project, including the module project.

Run the migration wizard in a test environment first

Because adapter migration might require you to change those applications that use version 7.5 of WebSphere Adapter for Lotus Domino, you must 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 7.5 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 IBM Integration Designer, complete the following steps.

The following steps describe how to run the adapter migration wizard from the connector project menu while in the Java EE perspective in IBM Integration Designer.

Procedure

1. Import the PI (project interchange) file for an existing project into the workspace.

Note: Ensure that you do not modify the contents of the RAR or copy the adapter JAR file outside the connector project.

2. When projects are created in an earlier version of IBM Integration Designer, the Workspace Migration wizard starts automatically and selects the projects to migrate. Follow the wizard and complete the workspace migration. For more information, see <http://publib.boulder.ibm.com/infocenter/dmndhelp/v7r5mx/topic/com.ibm.wbpm.wid.imuc.doc/topics/tmigscart.html>.
3. Change to the Java EE perspective.
4. Right-click the module and select **Migrate connector project**. For example, the adapter RAR module.

You can also launch the adapter migration wizard in the following ways:

- Right-click the project in the Java EE perspective and select **Migrate adapter artifacts**.
 - From the Problems view, right-click a migration-specific message and select **Quick Fix** to correct the problem.
5. In the Select Projects window, perform the following steps:
 - a. The **Source connector** field displays the name of the connector project that you are migrating. If you are migrating a module project, this field lists all the connector projects in the module project. Select the source project from the list. For more information, see “Migrating multiple adapters referred within a project” on page 31.
 - b. The **Target connector** field displays the name of the connector to which you are migrating. If you are working with more than one adapter version, this list displays the names of all the compatible connectors. Select the connector you want to migrate.
 - c. The **Target version** field displays the version corresponding to the target connector that you selected in the previous step.
 - d. The **Dependent artifacts project** area lists the adapter artifacts that are migrated. If you are migrating a module project, this area lists only the selected module project. If you are migrating a connector project within the module project, this area lists all projects which reference the selected connector project, including the module project. By default, all the dependent artifact projects are selected. If you do not select a dependent artifact project, that project is not migrated. You can migrate any project that you have not selected 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. For more information, see “Upgrading but not migrating a project” on page 33.

- e. Click **Next**. A warning window is displayed with the message, “Properties that are not supported in this version of the target adapter will be removed during the migration”.
 - f. Click **OK**.
6. In the Review Changes window, review the migration changes that occur in each of the artifacts that you are migrating. To view the details, expand each node by clicking the + sign.
 7. To complete the migration:
 - Click **Finish**.
 - If the files that need to be updated during migration are in read-only mode, you will be unable to click on the **Finish** button. To view these files, click **Next**. The Update Read-only files window displays the read-only files. To update these files and continue with the migration, click **Finish**. To exit the wizard without migrating the adapter, click **Cancel**.

Before running the migration process, the wizard performs a backup of 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.

Upon completing the migration successfully, all backed up projects are deleted.

8. If you are migrating an EAR file, optionally create a new EAR file with the migrated adapter and artifacts, and deploy it to IBM Business Process Manager or WebSphere Enterprise Service Bus. For more information about exporting and deploying an EAR file, see the topics devoted to it in this documentation.

Note: After completing the adapter and artifacts migration, update the WALD inbound extension manager to the latest version manually. The new extension manager files can be fetched from the <WID installation>/ResourceAdapters/<Domino adapter>/dependencies location. For information about configuring the WALD inbound extension manager, see “Deployment and configuration of the WALD inbound extension manager” on page 144.

Results

The project or EAR file is migrated to version 7.5. You do not need to run the external service wizard after exiting the adapter migration wizard.

Upgrading but not migrating a project

You can upgrade the adapter from an earlier version, to version 7.5 while choosing not to migrate the adapter project artifacts.

About this task

Running the migration wizard without selecting any adapter artifacts installs and upgrades your adapter. As the artifacts are not migrated, your applications cannot take advantage of the features and capabilities that exist in version 7.5 of the adapter.

Procedure

1. Import the PI (project interchange) file into the workspace.
2. When projects are created in an earlier version of IBM Integration Designer, the Workspace Migration wizard starts automatically and selects the projects to

migrate. Follow the wizard and complete the workspace migration. For more information, see <http://publib.boulder.ibm.com/infocenter/dmndhelp/v7r5mx/topic/com.ibm.wbpm.wid.imuc.doc/topics/tmigrscart.html>.

3. In the Java EE perspective, right-click the project name and click **Migrate connector project**. The **Adapter Migration** wizard is displayed.
4. In the Select Projects window, clear the dependent artifact projects, and click **Next**. A warning window is displayed with the message, "The properties that are not supported in the version of the target adapter will be removed during the migration."
5. Click **OK**.
6. In the Review Changes window, review the migration changes that occur during updating the project. To view the details, expand each node by clicking the + sign.
7. To complete the migration:
 - Click **Finish**.
 - If the files that need to be updated during migration are in read-only mode, you will be unable to click on the **Finish** button. To view these files, click **Next**. The Update Read-only files window displays the read-only files. To update these files and continue with the migration, click **Finish**. To exit the wizard without migrating the adapter, click **Cancel**.

Results

The project can now be used with WebSphere Adapter for Lotus Domino, version 7.5.

Chapter 3. Samples and tutorials

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

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

- From the welcome page of IBM Integration Designer, click **Go to Samples and Tutorials**. In the Samples and Tutorials pane, under More samples, click **Retrieve**. Browse the displayed categories to make your selection.
- From the Business Process Management Samples and Tutorials website:
<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 IBM Business Process Manager or WebSphere Enterprise Service Bus, use IBM Integration Designer to create a module, which is exported as an EAR file when you deploy the adapter.

Road map for configuring the module

Before you can use WebSphere Adapter for Lotus Domino in a run time 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 the adapter to use by using IBM Integration Designer. 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. See the topics following this road map for the details on how to perform each of the steps described here.

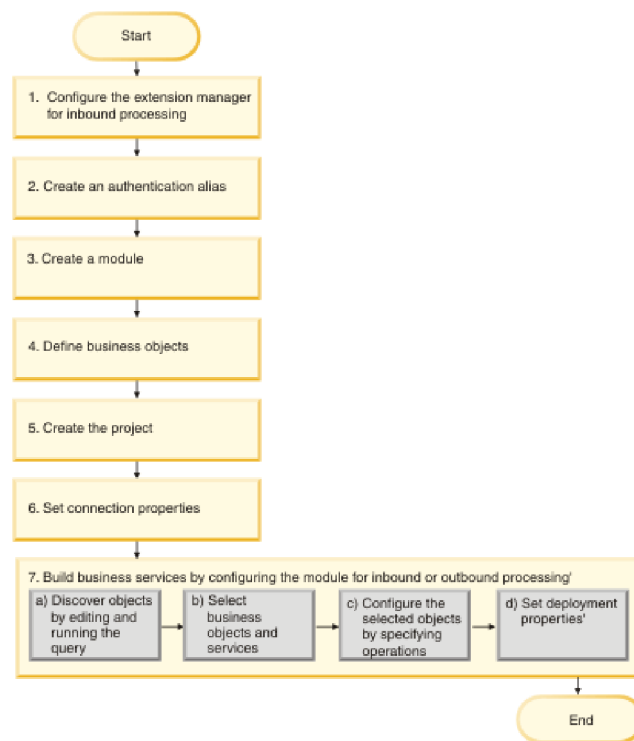


Figure 11. Road map for configuring the module

Configuring the module for deployment

This task consists of the following step:

1. Configure the Lotus Domino server to work with your adapter. You perform this step from within the Lotus Domino server.
2. Create an authentication alias to access the Lotus Domino server with an encrypted password. This step is optional, depending on your policy for handling passwords and IDs. You perform this step using the server.

3. Create the project. First, start the external service wizard in IBM Integration Designer to begin the process of creating and deploying a module. The wizard creates a project that is used to organize the files associated with the module.
4. Set connection properties that the external service wizard needs to connect to the Lotus Domino server for discovery of objects and services.
5. Configure the module for inbound or outbound processing by using the external service wizard to find and select business objects and services from the Lotus Domino server, and to generate business object definitions and related artifacts.
 - a. Select business objects and services for inbound or outbound processing from the business integration components discovered by the external service wizard.

Note: The required services on the domino server such as IIOP and HTTP must be running for both outbound and inbound processing. Also, ensure that the required files for inbound processing are copied in the correct directories.

- b. Configure the selected objects by specifying operations and other properties that apply to all the business objects.
- c. Set deployment properties that the adapter uses to connect to the Lotus Domino server at run time. Then, generate the service by using the external service wizard to save the new module, which contains the business object or objects you configured, the import or export file, and the service interface.

Creating an authentication alias

An authentication alias is a feature that encrypts the password used by the adapter to access the Lotus Domino server. The adapter can use it to connect to the Lotus Domino server instead of using a user ID and password stored in an adapter property.

Before you begin

To create an authentication alias, you must have access to the administrative console of IBM Business Process Manager or WebSphere Enterprise Service Bus. You must also know the user name and password to use to connect to the Lotus Domino server.

The following procedure describes how to access administrative console through IBM Integration Designer. If you are using the administrative console directly (without going through IBM Integration Designer), log in to the administrative console and skip to step 2 on page 39.

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.

Procedure

1. Start the administrative console:

To start the administrative console through IBM Integration Designer, perform the following steps:

- a. In the Business Integration perspective of Integration Designer, 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**. Wait until the status of the server is Started.
 - c. Right-click the name of the server and click **Run administrative console**.
 - d. Log on to the administrative console by clicking **Log in**. If your administrative console requires a user ID and password, type the ID and password first and 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. From 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 Lotus Domino server.
 - d. Optional: 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 specify when you configure the adapter properties later in the wizard.

Related concepts

“User authentication” on page 26

The adapter supports several methods for supplying the user name and password that are needed to connect to the Lotus Domino 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 IBM Integration Designer and a unit of deployment to IBM Business Process Manager or WebSphere Enterprise Service Bus. The module is packaged and deployed to IBM Business Process Manager or WebSphere Enterprise Service Bus as an enterprise archive (EAR) file.

Procedure

1. If IBM Integration Designer is not currently running, start it now.
 - a. Click **Start > Programs > IBM > IBM Integration Designer > IBM Integration Designer 7.5**.
 - b. If you are prompted to specify a workspace, either accept the default value or select another workspace.

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

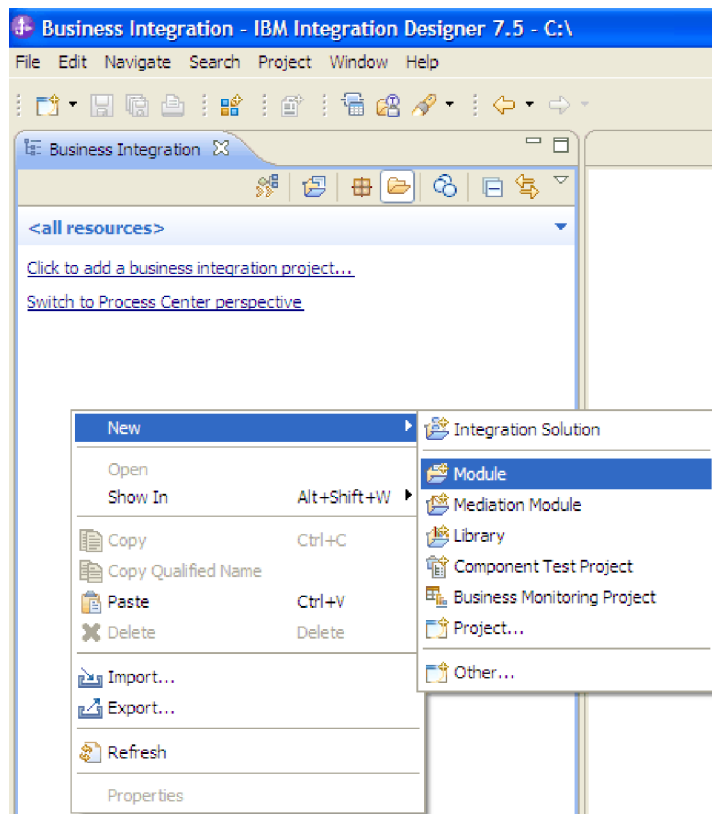


Figure 12. Business Integration section of the window

3. In the New Module window, type a name for the module in the **Module Name** field. For example, to configure a module for outbound processing, type Outbound as the module name. Click **Finish**.

Results

A new module is listed in the Business Integration window.

Creating the project

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

Before you begin

Ensure that you have gathered the information you need to establish a connection to the Lotus Domino server. For example, you need the name or IP address of the Lotus Domino server and the user ID and password to access it.

About this task

If you have an existing project, you can use it instead of creating a new one. Select it before you start the wizard.

Procedure

1. To start the external service wizard, go to the Business Integration perspective of IBM Integration Designer, and then click **File > New > External Service**. Click **Next**.
2. In the New External Service window, expand the **Adapters** folder, select **Domino**, and then click **Next**.
3. In the Select an Adapter window, select **IBM WebSphere Adapter for Lotus Domino (IBM : version)**, where *version* is the version of the adapter you want to use, and then Click **Next**.
4. In the Import a RAR File window, accept the default project name in the **Connector project** field or type a different name.
5. In the **Target runtime** field, select the type of server where you want to deploy the module. The wizard creates the artifacts that are appropriate to that server.
6. Click **Next**. The Locate the Required Files and Libraries window is displayed.

Results

A new connector project is created, which contains the adapter RAR file. The project is listed in the Business Integration perspective .

What to do next

Continue working in the external service wizard. The next step is to add database-specific files to the project.

Defining WebSphere Application Server environment variables

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

About this task

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

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 the following figure, the scope is defined at the node level as Node=S-XINWANGXNode04.

WebSphere Variables

WebSphere Variables

Use this page to define substitution variables. Variables specify a level of indirection as file system root directories. Variables have a scope level, which is either server, scope level can differ from values at other levels. When a variable has conflicting values, the value overrides values at greater scope levels. Therefore, server variables override cell variables, which override cell variables.

Scope: Cell=**S-XINWANGXNode03Cell**, Node=**S-XINWANGXNode04**

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 link](#)

Preferences

Maximum rows

Retain filter criteria

Show items at the following authorization group level:

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 `DOMINO_SERVER_IP` and the value is `127.0.0.1`. You can use the **Description** field, which is optional, to describe the purpose of the variable.

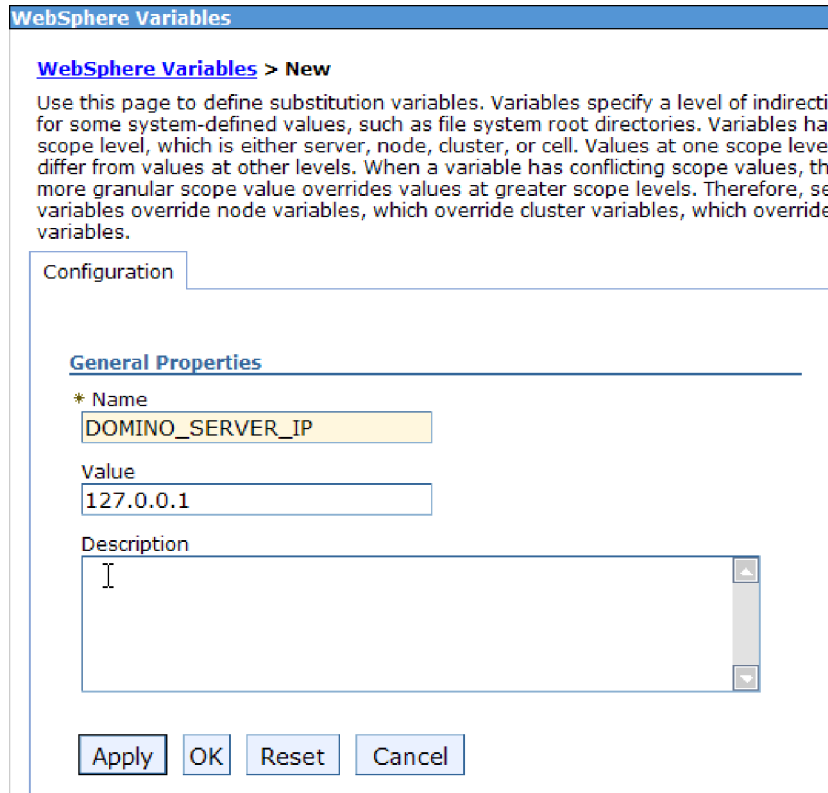


Figure 14. Providing a name and a value for the environment variable

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

Results

An environment variable called `DOMINO_SERVER_IP` is defined, with the value `127.0.0.1` and a scope of `Node=S-XINWANGXNode04`. You can use it in the external service wizard whenever you need to specify the run time Lotus Domino server host name.

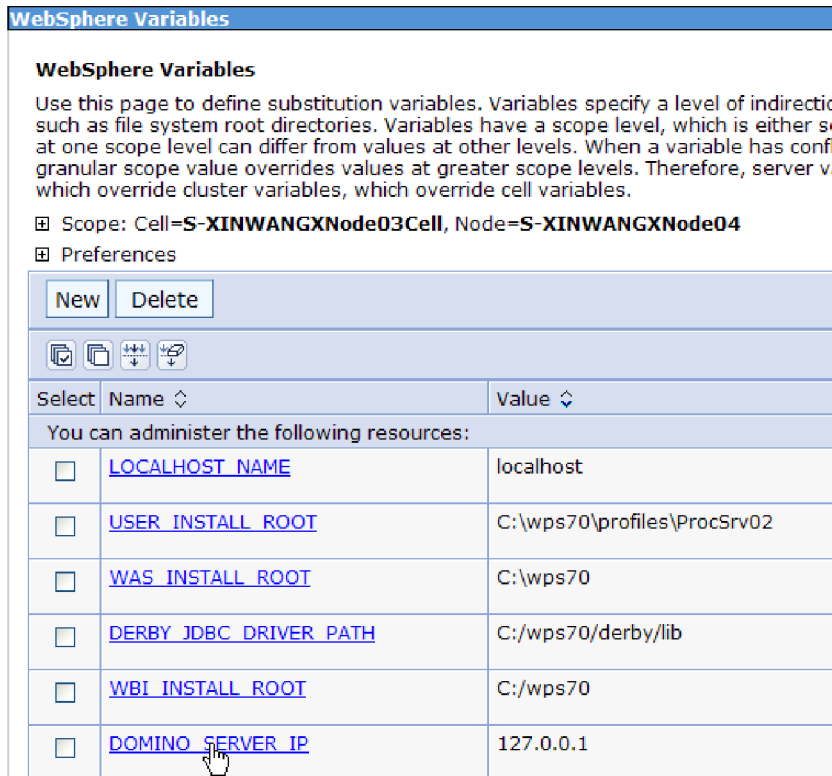


Figure 15. The new environment variable `DOMINO_SERVER_IP` is displayed in the WebSphere Variables window

Related concepts

“WebSphere Application Server environment variables” on page 22

WebSphere Application Server environment variables can be used in the external service wizard to specify directory values. You can now change any string property values in inbound and outbound configuration by only changing the environment variables.

Related reference

“Managed (J2C) connection factory properties” on page 121

Managed connection factory properties are used by the adapter at run time to create an outbound connection instance with the Lotus Domino server.

“Activation specification properties” on page 135

Activation specification properties are properties that hold the inbound event processing configuration information for a message endpoint.

Selecting the processing direction

The adapter processing direction can be configured at run time environment. You can either configure your adapter for inbound processing or outbound processing.

Before you begin

Ensure that you are running the external service wizard in IBM Integration Designer to perform this task.

About this task

When you configure your adapter for inbound processing you are processing the request or response interaction from the adapter to your service export. When you configure your adapter for outbound processing you are processing the response from your service import to the adapter.

In the Select the Processing Direction window, select **Outbound** or **Inbound**, and then click **Next** to advance to the Connection properties window.

Results

You have defined the type of process you need for your adapter.

Setting connection properties for the external service wizard

For the external service wizard to be able to discover business objects on the Lotus Domino server, you must specify the connection properties.

About this task

The external service wizard requires these properties to connect to the Lotus Domino server for discovery and for creating the service description.

Procedure

1. Start the external service wizard and enter the appropriate information in the Select the Processing Direction window, as explained in the “Selecting the processing direction” on page 44 topic.
2. In the Specify the Discovery Properties window, specify the connection properties for the wizard to connect to the Lotus Domino server. The connection information from this window is used by the adapter to connect to the Lotus Domino server and discover business objects. In this same wizard at a later time, you can also change the connection information or an authentication alias to be used at run time.
 - a. In the **Host name** field, specify the host name or IP address of the Lotus Domino server.
 - b. In the **DIOP port number** field, specify the port of the Lotus Domino server to be used.
 - c. In the **User name** field, enter the user name used to connect to the server.
 - d. In the **Password** field, enter the password used to connect to the server.
3. Optional: To specify the logging level, click **Change the logging properties for the wizard**, and then provide the following information:
 - In **Log file output location**, specify the location of the log file for the wizard.
 - In **Logging level**, specify the severity of errors that you want logged.

This log information is for the wizard only; at run time, the adapter writes messages and trace information into the standard log and trace files for the server.

4. Click **Next**.

Results

The external service wizard connects to the database and displays the Find Objects in the Enterprise System window.

What to do next

Continue working in the wizard. Locate the objects on the Lotus Domino server for which you want the wizard to create business objects.

Configuring the module for outbound processing

To configure a module to use the adapter for outbound processing, use the external service wizard in IBM Integration Designer to find and select business objects and services from the Lotus Domino server, and to generate the business object definitions and related artifacts.

Discovering and selecting objects

After connecting to the Domino server, you can run a query to search for the required databases. You can browse through the tree of discovered objects to understand the structure of objects in the Lotus Domino server, and also use filters to display only the database objects that you want to see.

Before you begin

You must have successfully connected to the Lotus Domino server. You must have a clear understanding of the data requirement of the program that needs to access the database.

About this task

This task starts in the Find Objects in the Enterprise System window of the external service wizard.

Procedure

1. In the Find Objects in the Enterprise System window, click **Edit Query** to modify the existing query as per your requirements. Type the database and form value to filter the databases and form matching the pattern. The database and form value take a wildcard pattern. The default value is *, where all databases or forms are displayed in the **Discovered objects** pane. The * and ? are valid wildcard characters, with * representing more than one character and ? representing a single character.

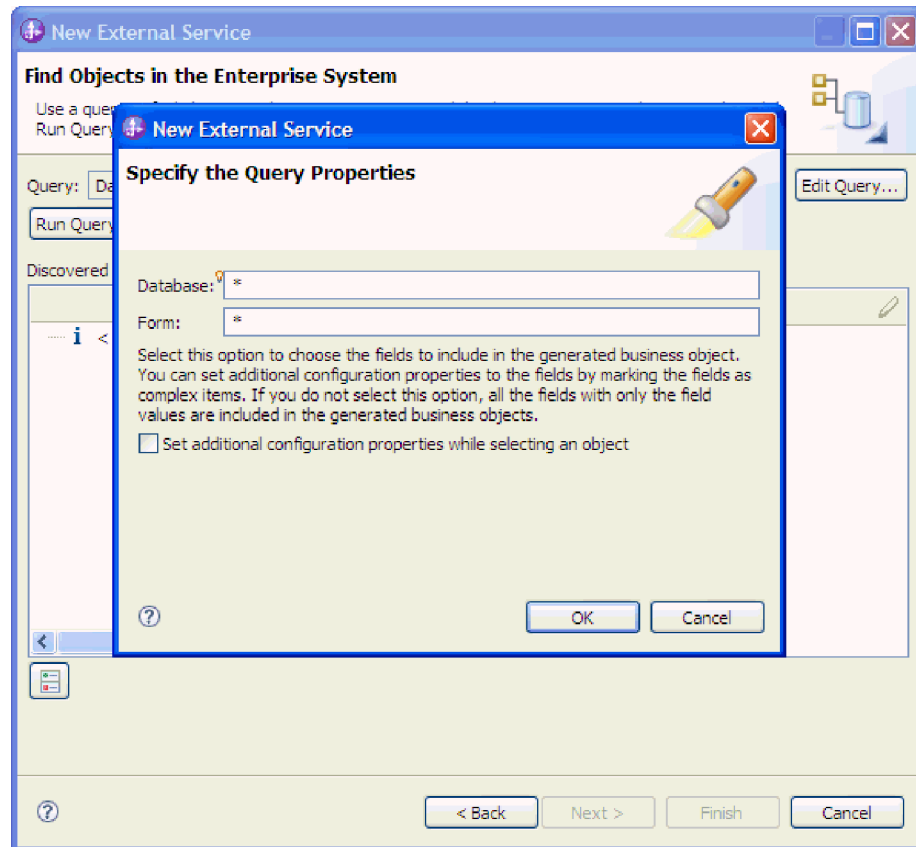


Figure 16. Specifying the query properties

2. Optional: **Configuring additional details for the business object:**
 - a. In the Specify the Query Properties window, select the **Set additional configuration properties while selecting an object** check box.

Select this check box to configure additional details when you move a form to the selected objects pane.

3. Click **OK** after entering all the data in the Specify the Query Properties window.
4. Click **Run Query**. The databases matching the pattern entered in the query properties window is displayed in the **Discovered objects** pane.

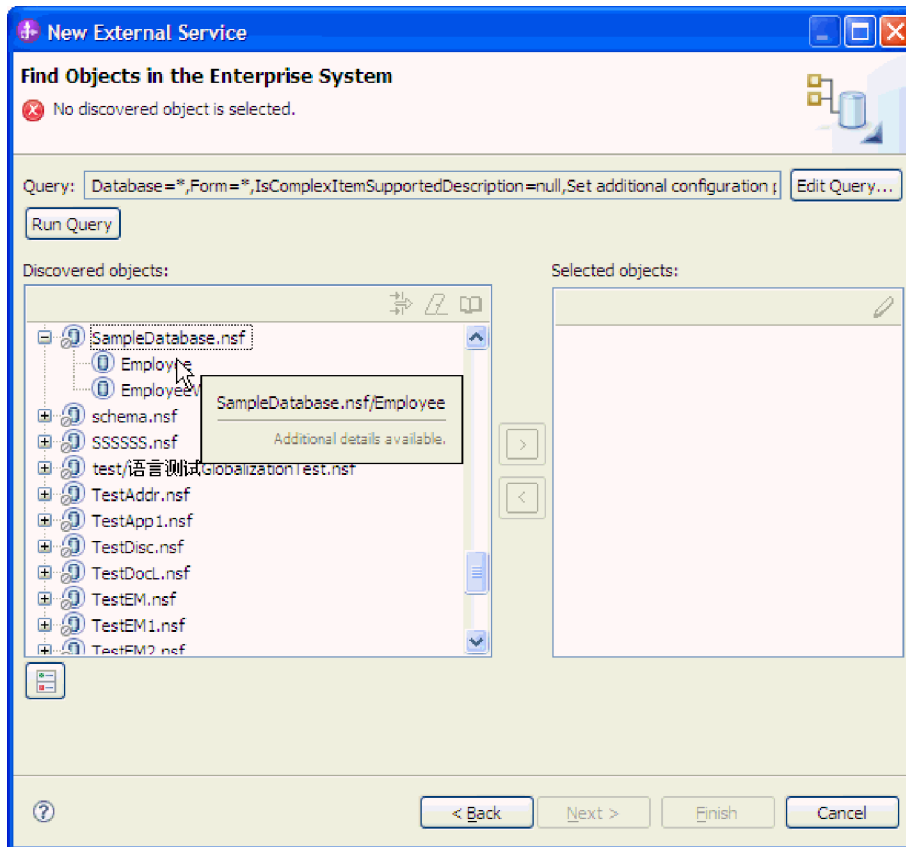
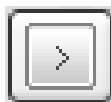


Figure 17. Specifying the configuration properties

Note: To restore an expired database connection, restart the external service wizard.

5. In the **Discovered objects** list, click + (the plus sign) to expand a node, to see the form names discovered by the wizard.
6. Select the required objects from the **Discovered objects** listing and use the



(Add) button to add them to the **Selected objects** listing.

7. Optional: If you are configuring additional details for the business object, in the Configuration Properties window, you need to run this step. The Configuration Properties window is displayed only if **Set additional configuration properties while selecting an object** check box in the Specify the Query Properties window is selected and a form name from **Discovered objects** is moved to **Selected objects**.

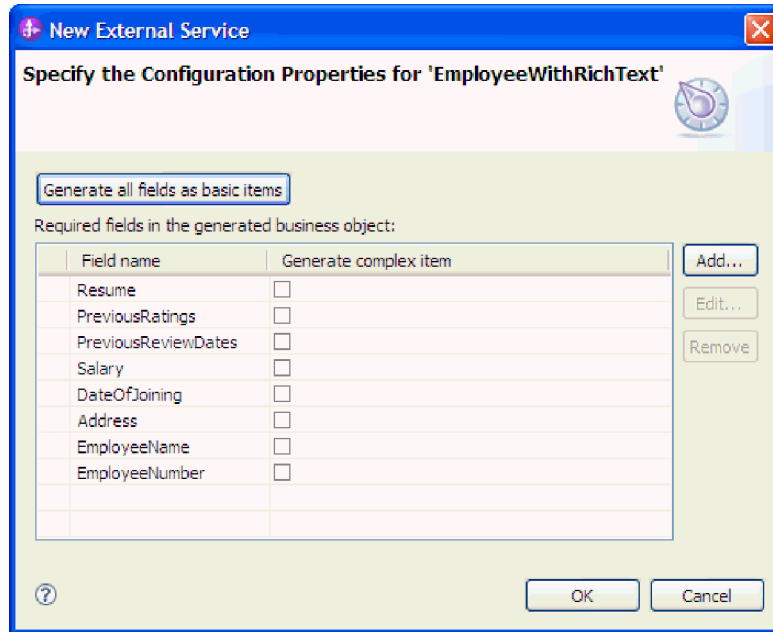


Figure 18. Setting additional configuration properties while selecting an object

- a. Select the form fields which you want to generate in the business object schema. This step is helpful in cases where a form contains many fields but your integration scenario needs only few fields. Click **Remove** to remove the fields that are not required. You can also add the field again by clicking **Add**. By default, all the form fields are generated in the business object schema.
 - b. Select the **Generate complex item** check boxes for the form fields for which you need more information. During run time, more detailed information (information about **IsSummary**, **IsEncrypted**, **IsReaders**) is displayed for each item. For more information about the business object structure see, Table 8 on page 113.
 - c. Click **Generate all fields as basic items** to reset the values to default.
8. Click **Next**.

Results

The wizard displays the document objects you have selected to be used by the adapter.

What to do next

Continue working in the external service wizard. The next step is to select the objects you want to use in your module, configure each business object, and create hierarchies of business objects.

Related concepts

“Business object structures” on page 17

A business object is a structure that consists of data and additional instructions, if any, for processing the data. The adapter supports the form-based business object structure. A form-based type is a content-specific business object that supports specific business object structures (such as, create and retrieve business objects).

“The external service wizard” on page 20

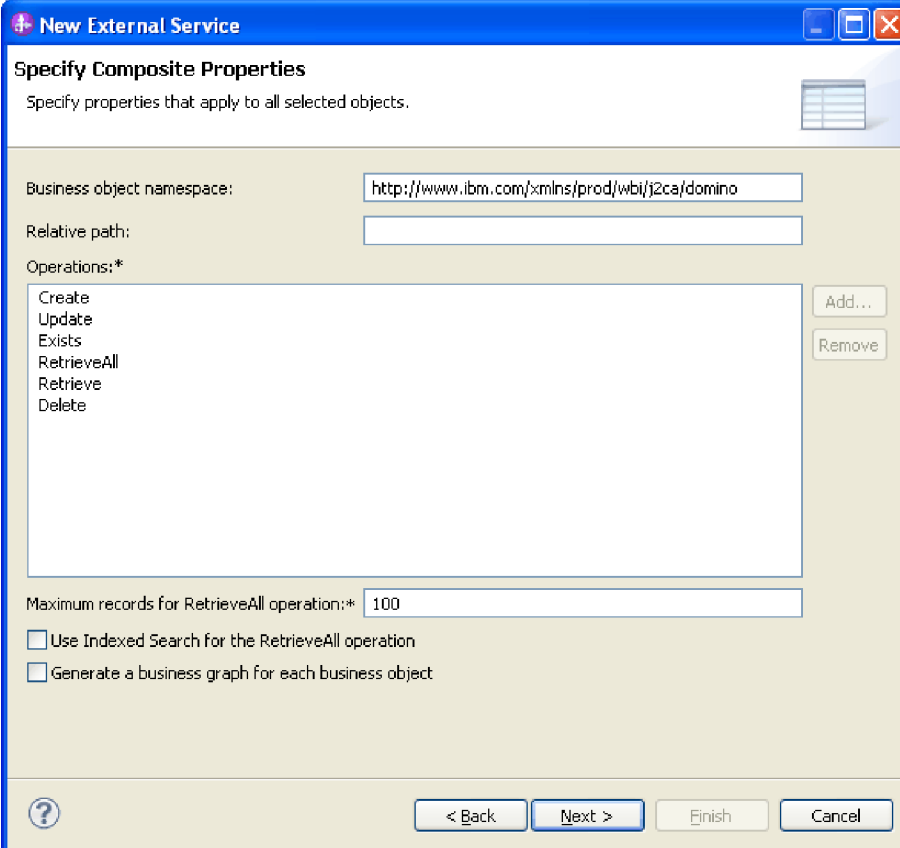
The external service wizard in WebSphere Adapter for Lotus Domino is used to create services and to generate business objects from the selected objects. The wizard also generates the module and the service artifacts that enable the adapter to run as a Service Component Architecture (SCA) component.

Configuring the selected objects

After you have selected metadata object in the external service wizard, you need to specify properties that apply to them.

Procedure

1. When the **Selected objects** list in the Find Objects in the Enterprise System window contains all the business objects you want to use in your application, click **Next**.
2. In the Specify Composite Properties window, the supported outbound operations are listed in the **Operations** field. The operations available for you to configure are: Create, Update, Delete, Exists, Retrieve, and RetrieveAll.



The screenshot shows a Windows-style dialog box titled "New External Service" with a sub-title "Specify Composite Properties". The main instruction is "Specify properties that apply to all selected objects." The form contains the following fields and controls:

- Business object namespace:** A text box containing the URL "http://www.ibm.com/xmlns/prod/wbi/j2ca/domino".
- Relative path:** An empty text box.
- Operations:*** A list box containing the following items: Create, Update, Exists, RetrieveAll, Retrieve, and Delete. To the right of the list are "Add..." and "Remove" buttons.
- Maximum records for RetrieveAll operation:*** A text box containing the value "100".
- Two checkboxes: "Use Indexed Search for the RetrieveAll operation" and "Generate a business graph for each business object", both of which are currently unchecked.
- At the bottom, there are four buttons: a help icon (?), "< Back", "Next >", "Finish", and "Cancel".

Figure 19. Specifying the composite properties

3. To configure an operation, you can use either the **Add** or the **Remove** option. If you remove any of these operations, the removed operation is made available in the Add Value window for you to add back. Use the **Add** option again to open the Add Value window and select the operations to be added back.
4. In the **Business object namespace** field, either accept the default namespace or type the full name of another namespace.
The namespace is prefixed to the business object name to keep the business object schemas logically separated.
5. Optional: In the **Relative path** field, type the relative path to the folder where the generated business objects are to be stored. Click **Next**.

Note: If you are creating multiple adapter artifacts within a module, ensure that you specify different business object folders for each adapter within the module. For example, if you are creating artifacts for Oracle, JDBC, SAP, and JDE within a module, you need to create different relative folders for each of these adapters. If you do not specify different relative folders, the existing artifacts are overwritten when you generate new artifacts.

6. In the **Maximum records for RetrieveAll operation** field, type the upper limit on the number of records to retrieve for a RetrieveAll operation. The default value is 100. For more information about this property, see “Interaction specification properties” on page 127.
7. Select the **Use Indexed Search for the RetrieveAll operation** option, to specify the index-based text searches.

Note: If you remove the RetrieveAll operation from the **Operations** list, the **Maximum records for RetrieveAll operation** and the **Use Indexed Search for the RetrieveAll operation** fields are disabled.

Results

You have provided information that applies to all business objects in the module.

What to do next

Continue working in the wizard. The next step is to specify deployment information to use at run time and information for saving the service as a module.

Related concepts

“Business object structures” on page 17

A business object is a structure that consists of data and additional instructions, if any, for processing the data. The adapter supports the form-based business object structure. A form-based type is a content-specific business object that supports specific business object structures (such as, create and retrieve business objects).

“The external service wizard” on page 20

The external service wizard in WebSphere Adapter for Lotus Domino is used to create services and to generate business objects from the selected objects. The wizard also generates the module and the service artifacts that enable the adapter to run as a Service Component Architecture (SCA) component.

Setting deployment properties and generating the service

After you select and configure business objects for your module, use the external service wizard to configure properties that the adapter uses to connect to a Lotus Domino server system. The wizard creates a new business integration module where all the artifacts and property values are saved. This task is performed using

the Specify the Service Generation and Deployment Properties and Specify the Location Properties windows of the external service wizard.

Related reference

“Interaction specification properties” on page 127

Interaction specification properties control the interaction for an operation. The external service wizard sets the interaction specification properties when you configure the adapter. Typically, you do not need to change these properties. However, you can change some properties for outbound operations. To change these properties after the application is deployed, use the assembly editor in IBM Integration Designer. The properties are located in the method binding of the import.

Setting deployment properties for a runtime environment

The connection properties are initialized to the values that the wizard used to connect to the database. To configure the module to use other values, change values such as user name and password here.

Before you begin

Ensure that you have configured the business object.

About this task

To generate artifacts for your module follow this procedure:

Procedure

1. In the Specify the Service Generation and Deployment Properties window, click **Edit Operations** to review the names of or add a description for the operations for the business objects that you are creating.
2. In **Deployment Properties**, specify how you want the adapter to connect to the Lotus Domino server by selecting any one of these authentication methods.
 - To use a J2C authentication alias, click **Using an existing JAAS alias (recommended)** and type the name of the alias in the **J2C authentication data entry** field. You can either specify an existing authentication alias or create one at any time before deploying the module. The name is case sensitive and includes the node name.
 - To use the security properties from the managed connection properties, click **Using security properties from the managed connection factory** and enter the following information:
 - In the **User name** field, type the user name to connect to the Domino server.
 - In the **Password** field, type the password to connect to the Domino server.

Note: The security properties are not encrypted and stored as plain text.

 - To use any other method of authenticating to the Domino server, click **Other**.
3. In the **Deploy connector project** field, specify whether to include the adapter files in the module. Select 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.
4. If you selected **On server for use by multiple adapters** in the previous step, specify how you want to set the connection properties at runtime.
- If you manually created and configured a managed connection factory or activation specification on the server or if you have already deployed an application that connects to the same Lotus Domino server using the same managed connection factory or activation specification properties, you can reuse the managed connection factory or activation specification by specifying the name of its Java Naming and Directory Interface (JNDI) data source:
 - a. In **Connection properties**, select **Use predefined connection properties**.
 - b. In **JNDI Lookup Name**, type the name of the JNDI data source for an existing managed connection factory or activation specification.
 - c. Click **Next**.
 - If this application is the first one that connects to the database with a specific user name and password, or if you want to administer the user name and password separately from other applications, select **Specify connection properties**.

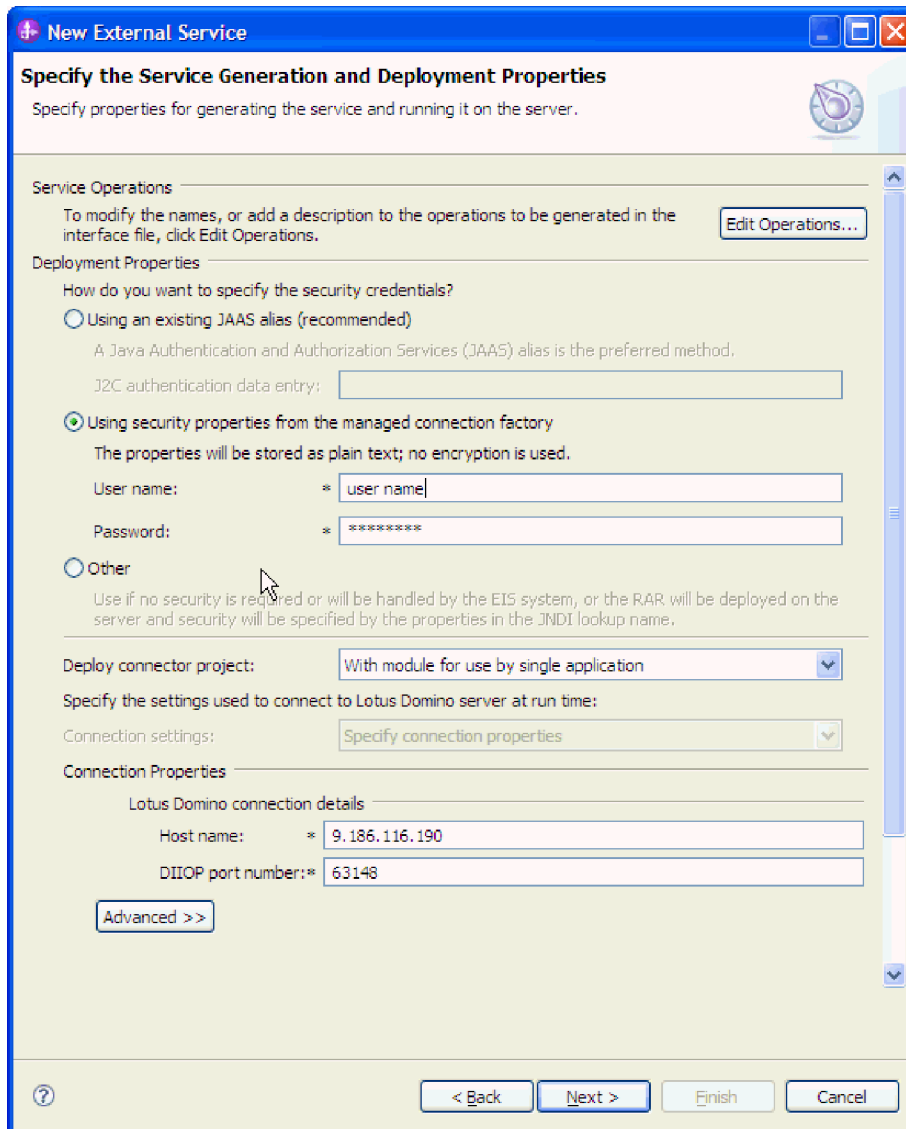


Figure 20. Specify the service generation and deployment properties window

5. Optional: Review and if necessary, change the values of the required connection properties. The fields are initialized with the connection information you specified when you started the wizard.
 - **Connection properties**
 - In the **Host name** field, type the host name or IP address of the Lotus Domino server.
 - In the **DIOP port number** field, type the port number of the Lotus Domino server. The default DIOP port number is 63148.
6. Click **Advanced**. Expand the advanced sections to review and specify the following options:
 - **Processing methods for illegal XML characters**
 - a. From the **Illegal XML character processing** drop-down list, select one of the options that correspond to your requirement:
 - **Do not validate illegal XML characters**, if you want to work with the default adapter behavior.

- **Throw exception if contents in the BO contains illegal XML characters**, if you want to proceed after you get an exception message, and log the illegal XML characters in the trace file, at runtime.
- **Discard all illegal XML characters and related logs**, if you want the adapter to discard the illegal XML characters, and log the characters in the trace file, at runtime.

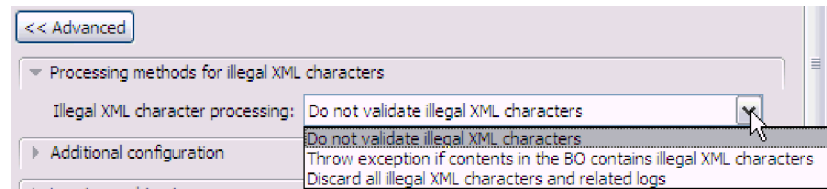


Figure 21. Processing methods for illegal XML characters

- **Additional configuration**

- a. To specify support for business faults, select one of the choices from the **Support for business faults during runtime** list.
 - To enable fault handling only for the business faults, accept the default setting **Only business faults (1)**.
 - To specify fault handling support for both business and WebSphere Adapter for Lotus Domino faults, select the **Business faults and Lotus Domino faults (2)** option from the list.
 - To completely disable fault handling support, select the **No fault handling and log the errors as exception (0)** option from the list. For more information, see “Set business faults behavior during run time ” on page 126.
- b. To specify the number of connection exceptions during the outbound operation, set **Maximum number of retries in case of system connection failure** to a value greater than or equal to 0. For more information, see “Maximum retries on connection failure” on page 124.
- c. To specify the time interval between attempts to connect to the Lotus Domino server if a connection fails, set **Connection retry interval** to a value in milliseconds. For more information, see “Retry interval if connection fails (in milliseconds) entities” on page 125.

- **Logging and tracing**

- a. If you have multiple instances of the adapter, set Adapter ID to a value that is unique for this instance. For more information, see “Adapter ID” on page 122.
- b. You can select the **Disguise user data as 'XXX' in log and trace files** check box to prevent sensitive user data from being written to log and trace files. For more information, see “Disguise user data as "XXX" in log and trace files” on page 123.

7. Click **Next**.

Results

The Specify the Location Properties window is displayed.

Related concepts

“Outbound processing” on page 3

WebSphere Adapter for Lotus Domino supports outbound request processing. When the adapter receives a request, which is sent in the form of a business object from the module, it processes the request and returns the result as a business object. The adapter provides a standard set of outbound operations.

“User authentication” on page 26

The adapter supports several methods for supplying the user name and password that are needed to connect to the Lotus Domino 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 (J2C) connection factory properties” on page 121

Managed connection factory properties are used by the adapter at run time to create an outbound connection instance with the Lotus Domino server.

“Resource adapter properties” on page 119

The resource adapter properties control the general operation of the adapter, such as specifying the namespace for business objects. You set the resource adapter properties using the external service wizard when you configure the adapter. After deploying the adapter, use the administrative console to change these properties.

“Interaction specification properties” on page 127

Interaction specification properties control the interaction for an operation. The external service wizard sets the interaction specification properties when you configure the adapter. Typically, you do not need to change these properties. However, you can change some properties for outbound operations. To change these properties after the application is deployed, use the assembly editor in IBM Integration Designer. The properties are located in the method binding of the import.

“Connection properties for the wizard” on page 117

External service connection properties are used to establish a connection between the external service wizard, a tool that is used to create business objects, and the Lotus Domino server. These properties specify such things as connection configuration, bidirectional transformation properties, and logging options for the wizard. Once a connection is established, the wizard can discover in the Lotus Domino server the metadata it needs to create business objects.

“Globalization” on page 150

WebSphere Adapter for Lotus Domino 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.

Setting location properties for artifacts

After you select and configure deployment properties, use the external service wizard to configure location properties that the adapter uses to generate the service.

Before you begin

Ensure that you have configured the business object.

About this task

To configure location properties for your module follow this procedure:

Procedure

1. In the Specify the Location Properties window, select an existing module or create a new one.
 - To use an existing module, select its name from the **Module** list.

Important: If the module contains an interface or business object with the same name as that you are now configuring, the original interface or business object in the module is replaced by the new version.
 - To create a new module:
 - a. Click **New**.
 - b. In the Select a Business Integration Project Type window, select **Module** and click **Next**.
 - c. In the Create a Module window, type a name for the module in the **Module name** field. An example of a module name is DominoOutboundInterface.
 - d. To retain the service description files (the `.import` and the `.wsdl` files) to be located in the default folder in the module, leave the **Use default location** check box selected. To specify a different folder in the module, clear the **Use default location** check box, and then click **Browse** to specify a different folder in the **Location** field.
 - e. To automatically open the module in the assembly diagram in Integration Designer when the wizard closes, select the **Open the module assembly diagram** check box.
 - f. Click **Finish** to create the module.
2. Specify the namespace you want to use for your outbound interface.
 - Select **Use the default namespace** check box to retain the outbound interface to use the default derived namespace. By default, this check box is selected.
 - To specify a different namespace, clear **Use the default namespace** check box and type a different value in the **Namespace** field.
3. Specify the folder in the new module where the service description is to be saved. In the **Folder** field, type the folder name or browse to an existing folder. If you do not specify a folder name, the artifacts (the `import`, `XSD`, and `WSDL` files) are stored in the root folder of the module (the folder with the module name).
4. In the **Name** field, accept the default import name or type a different name.
5. If you want to save business objects in a library where they can be used by other modules, select **Save business objects to a library** check box and specify the location of the library in the **Library** field.
6. In the **Description** field, type a descriptive comment about the module.
7. Click **Finish** to complete the setting of location properties.

Note: If the Model Changed window is displayed, click **Yes**.

Results

The wizard exits. The artifacts are generated.

Related concepts

“Outbound processing” on page 3

WebSphere Adapter for Lotus Domino supports outbound request processing. When the adapter receives a request, which is sent in the form of a business object from the module, it processes the request and returns the result as a business object. The adapter provides a standard set of outbound operations.

Related reference

“Managed (J2C) connection factory properties” on page 121

Managed connection factory properties are used by the adapter at run time to create an outbound connection instance with the Lotus Domino server.

“Resource adapter properties” on page 119

The resource adapter properties control the general operation of the adapter, such as specifying the namespace for business objects. You set the resource adapter properties using the external service wizard when you configure the adapter. After deploying the adapter, use the administrative console to change these properties.

“Interaction specification properties” on page 127

Interaction specification properties control the interaction for an operation. The external service wizard sets the interaction specification properties when you configure the adapter. Typically, you do not need to change these properties. However, you can change some properties for outbound operations. To change these properties after the application is deployed, use the assembly editor in IBM Integration Designer. The properties are located in the method binding of the import.

“Connection properties for the wizard” on page 117

External service connection properties are used to establish a connection between the external service wizard, a tool that is used to create business objects, and the Lotus Domino server. These properties specify such things as connection configuration, bidirectional transformation properties, and logging options for the wizard. Once a connection is established, the wizard can discover in the Lotus Domino server the metadata it needs to create business objects.

“Globalization” on page 150

WebSphere Adapter for Lotus Domino 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.

Configuring the module for inbound processing

To configure a module to use the adapter for inbound processing, use the external service wizard in IBM Integration Designer to find and select business objects and services from the Lotus Domino server, and to generate business object definitions and related artifacts.

Discovering and selecting objects

After connecting to the Lotus Domino server, you can run a query to search for the required databases. You can browse through the tree of discovered objects to understand the structure of objects in the Lotus Domino server, and also use filters to display only the database objects that you want to see.

Before you begin

You must have successfully connected to the Lotus Domino server. You must have a clear understanding of the data requirement of the program that needs to access the database.

About this task

This task starts in the Find Objects in the Enterprise System window of the external service wizard.

Procedure

1. In the Find Objects in the Enterprise System window, click **Edit Query** to modify the existing query based on your requirements. Type the database and form value to filter the databases and form matching the pattern. The database and form value take a wildcard pattern. The default value is *, where all databases or forms are displayed in the **Discovered objects** pane. The * and ? are valid wildcard characters, with * representing more than one character and ? representing a single character.

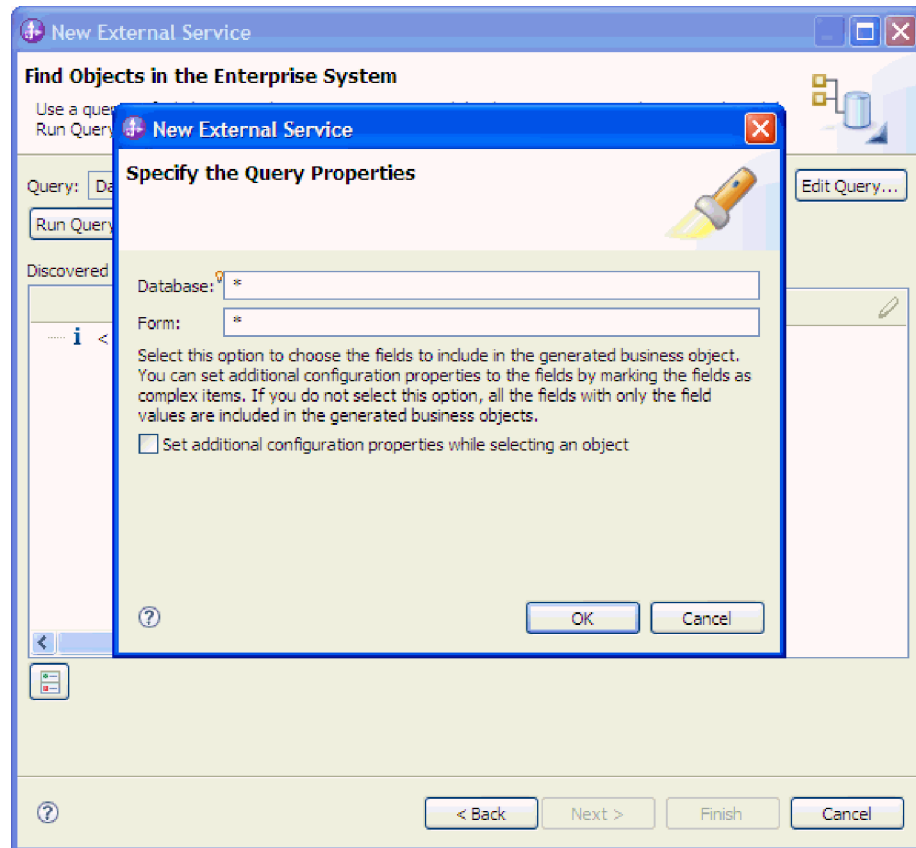


Figure 22. Specifying the query properties

2. Optional: **Configuring additional details for the business object:**
 - a. In the Specify the Query Properties window, select the **Set additional configuration properties while selecting an object** check box.

Select this check box to configure additional details when you move a form to the selected objects pane.

3. Click **OK** after entering all the data in the Specify the Query Properties window.
4. Click **Run Query**. The databases matching the pattern entered in the Specify the Query Properties window is displayed in the **Discovered objects** pane.

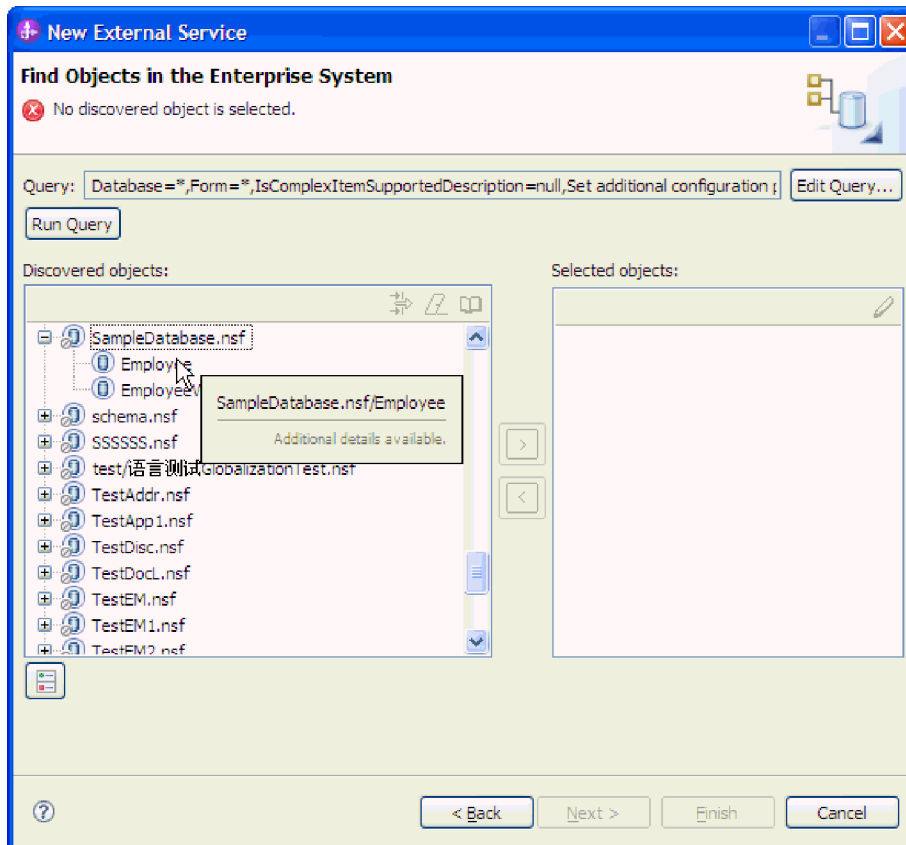


Figure 23. Specifying the configuration properties

Note: To restore an expired database connection, restart the external service wizard.

5. In the **Discovered objects** list, click + (plus sign) to expand a node, to see the form names discovered by the wizard.
6. Select the required objects from the **Discovered objects** listing and click the



(Add) button to add them to the **Selected objects** listing.

7. Optional: If you are configuring additional details for the business object, in the Specify the Configuration Properties for 'object' window, you need to run this step. The Specify the Configuration Properties for 'object' window is displayed only if **Set additional configuration properties while selecting an object** check box in the Specify the Query Properties window is selected and a form name from **Discovered objects** is moved to **Selected objects**.

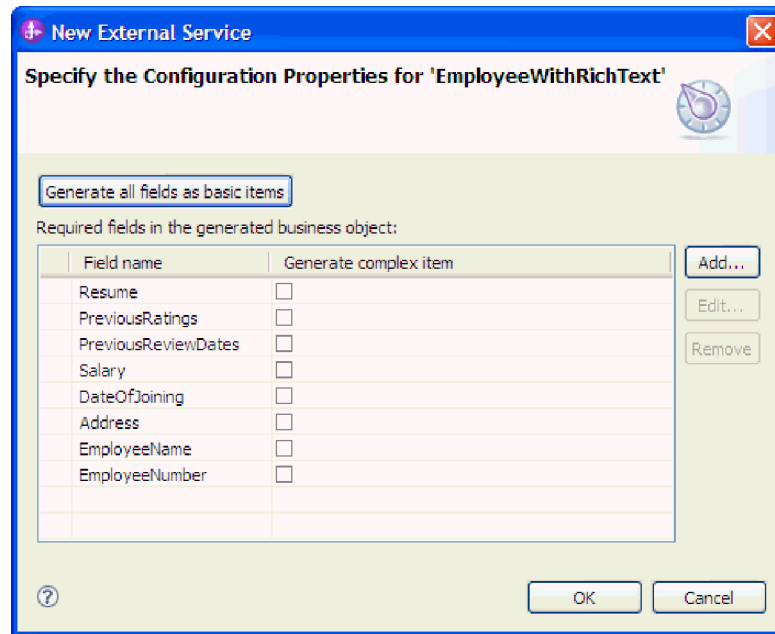


Figure 24. Setting additional configuration properties while selecting an object

- a. Select the form fields which you want to generate in the business object schema. You can use this configuration when a form has many fields but in the integration scenario you need only few of those fields. Click **Remove** to remove the fields that are not required. You can also add the field again by clicking **Add**. By default all the form fields are generated in the business object schema.
- b. Select the **Generate complex item** check boxes for the form fields that you need more information. During run time, more detailed information (information about **IsSummary**, **IsEncrypted**, **IsReaders**) is displayed for each item. For more information about the business object structure see, Table 8 on page 113.
- c. Click **Generate all fields as basic items** to reset the values to default.

Results

The wizard displays the document objects you have selected to be used by the adapter.

What to do next

Continue working in the external service wizard. The next step is to select the objects you want to use in your module, configure each business object, and create hierarchies of business objects.

Related concepts

“Business object structures” on page 17

A business object is a structure that consists of data and additional instructions, if any, for processing the data. The adapter supports the form-based business object structure. A form-based type is a content-specific business object that supports specific business object structures (such as, create and retrieve business objects).

“The external service wizard” on page 20

The external service wizard in WebSphere Adapter for Lotus Domino is used to create services and to generate business objects from the selected objects. The wizard also generates the module and the service artifacts that enable the adapter to run as a Service Component Architecture (SCA) component.

Configuring the selected objects

After you have selected metadata object in the external service wizard, you need to specify properties that apply to them.

Procedure

1. When the **Selected objects** list in the Find Objects in the Enterprise System window contains all the business objects you want to use in your application, click **Next**.
2. In the Specify Composite Properties window, the supported inbound operations are listed in the **Operations** field. The operations available for you to configure are: Create, Update, and Delete.

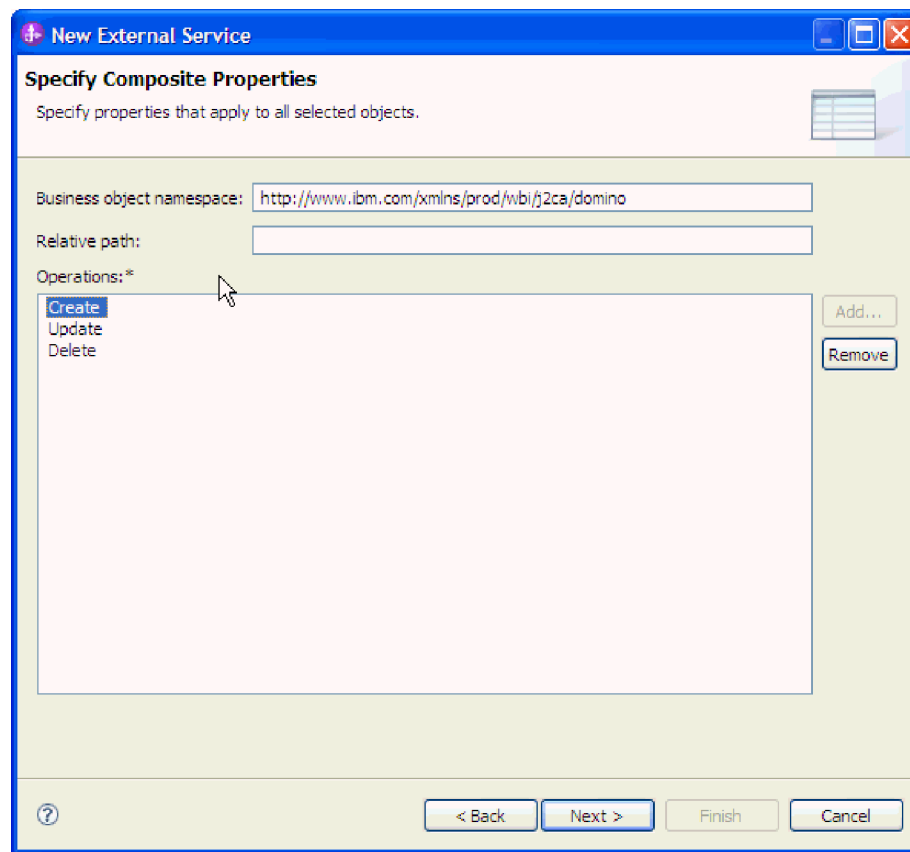


Figure 25. Specifying the composite properties

3. To configure an operation, you can use either the **Add** or the **Remove** option. If you remove any of these operations, the removed operation is made available in the Add Value window for you to add back. Use the **Add** option again to open the Add Value window and select the operations to be added back.
4. In the **Business object namespace** field, either accept the default namespace or type the full name of another namespace.
The namespace is prefixed to the business object name to keep the business object schemas logically separated.
5. Optional: In the **Relative path** field, type the relative path to the folder where the generated business objects are to be stored. Click **Next**.

Note: If you are creating multiple adapter artifacts within a module, ensure that you specify different business object folders for each adapter within the module. For example, if you are creating artifacts for Oracle, JDBC, SAP, and JDE within a module, you need to create different relative folders for each of these adapters. If you do not specify different relative folders, the existing artifacts are overwritten when you generate new artifacts.

Results

You have provided information that applies to all business objects in the module.

What to do next

Continue working in the wizard. The next step is to specify deployment information to use at run time and information for saving the service as a module.

Related concepts

“Business object structures” on page 17

A business object is a structure that consists of data and additional instructions, if any, for processing the data. The adapter supports the form-based business object structure. A form-based type is a content-specific business object that supports specific business object structures (such as, create and retrieve business objects).

“The external service wizard” on page 20

The external service wizard in WebSphere Adapter for Lotus Domino is used to create services and to generate business objects from the selected objects. The wizard also generates the module and the service artifacts that enable the adapter to run as a Service Component Architecture (SCA) component.

Setting deployment properties and generating the service

After you select and configure business objects for your module, use the external service wizard to configure properties that the adapter uses to connect to a Lotus Domino server system. The wizard creates a business integration module where all the artifacts and property values are saved. This task is performed using the Specify the Service Generation and Deployment Properties and Specify the Location Properties windows of the external service wizard.

Setting deployment properties for a runtime environment

The connection properties are initialized to the values that the wizard used to connect to the database. To configure the module to use other values, change values such as user name and password here.

Before you begin

This task is performed using the Specify the Service Generation and Deployment Properties window of the external service wizard.

About this task

To generate artifacts for your module follow this procedure:

Procedure

1. In the Specify the Service Generation and Deployment Properties window, click **Edit Operations** to review the names or add a description for the operations for the business objects that you are creating.
2. In **Deployment Properties**, specify how you want the adapter to connect to the Lotus Domino server by selecting any one of these authentication methods.
 - To use a J2C authentication alias, click **Using an existing JAAS alias (recommended)** and type the name of the alias in the **J2C authentication data entry** field. You can either specify an existing authentication alias or create one at any time before deploying the module. The name is case-sensitive and includes the node name.
 - To use the security properties from the managed connection properties, click **Using security properties from the managed connection factory** and enter the following information:
 - In the **User name** field, type the user name to connect to the Lotus Domino server.
 - In the **Password** field, type the password to connect to the Lotus Domino server.

Note: The security properties are not encrypted and stored as plain text.

 - To use any other method of authenticating to the Lotus Domino server, click **Other**.
3. In the **Deploy connector project** field, specify whether to include the adapter files in the module. Select 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.
4. If you selected **On server for use by multiple adapters** in the previous step, specify how you want to set the connection properties at run time.
 - If you manually created and configured a managed connection factory or activation specification on the server or if you have already deployed an application that connects to the same Lotus Domino server using the same managed connection factory or activation specification properties, you can

reuse the managed connection factory or activation specification by specifying the name of its Java Naming and Directory Interface (JNDI) data source:

- a. In **Connection properties**, select **Use predefined connection properties**.
 - b. In **JNDI Lookup Name**, type the name of the JNDI data source for an existing managed connection factory or activation specification.
 - c. Click **Next**.
- If this application is the first one that connects to the database with a specific user name and password, or if you want to administer the user name and password separately from other applications, select **Specify connection properties**.

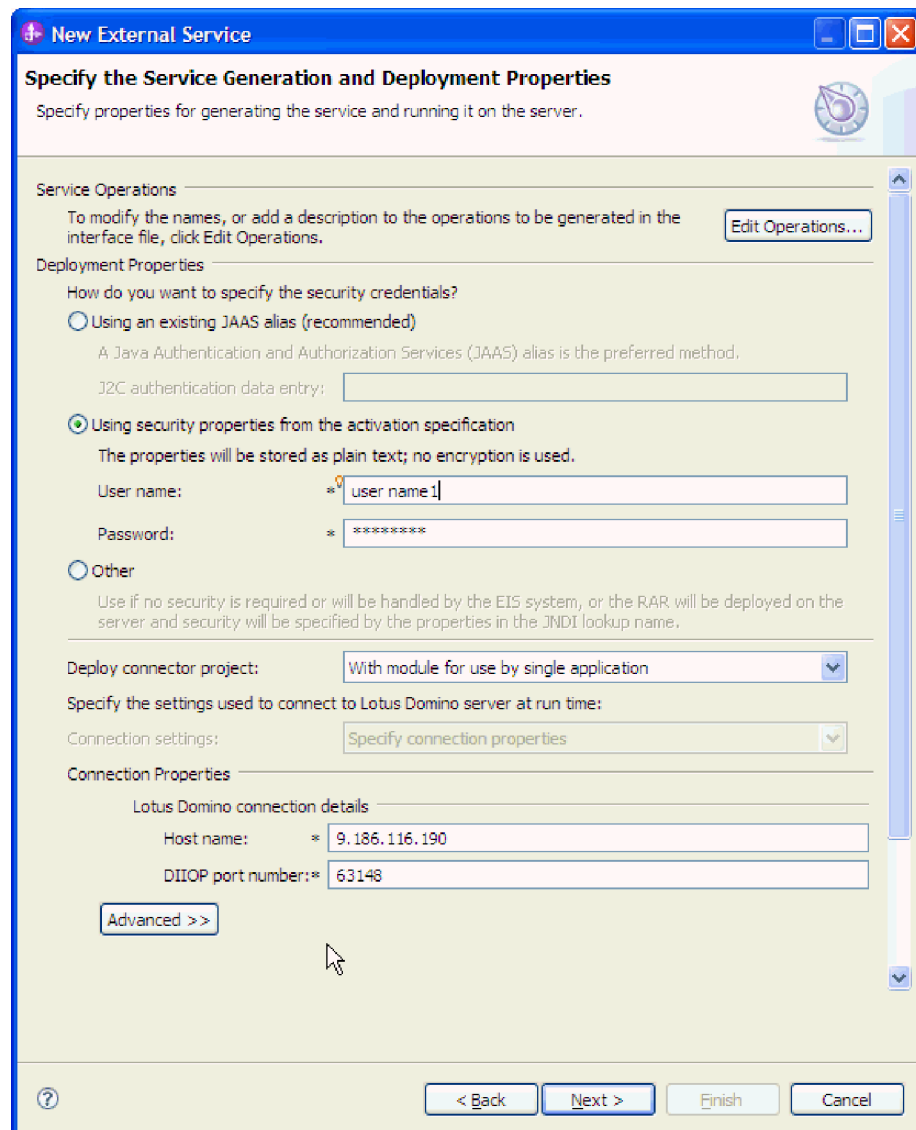


Figure 26. Specify the service generation and deployment properties window

5. Optional: Review and if necessary, change the values of the required connection properties. The fields are initialized with the connection information you specified when you started the wizard.
 - **Connection properties**

- In the **Host name** field, type the host name or IP address of the Lotus Domino server.
 - In the **DIIO port number** field, type the port number of the Lotus Domino server. The default DIIO port number is 63148.
6. Click **Advanced**. Expand the advanced sections to review and specify the following options:
- **Event polling configuration**
 - a. In the **Interval between polling periods (milliseconds)** field, type the time in milliseconds, that the adapter waits between polling periods. For more information about the property, see “Interval between polling periods (PollPeriod)” on page 140.
 - b. In the **Maximum events in polling period** field, type the number of events to deliver in each polling period. For more information about the property, see “Maximum events in polling period (PollQuantity)” on page 140.
 - c. In the **Time between retries in case of case of system connection failure (in milliseconds)** field, type the time in milliseconds, that the adapter waits before trying to connect after a connection failure during polling. For more information, see “Retry interval if connection fails (RetryInterval)” on page 142.
 - d. In the **Maximum number of retries in case of system connection failure** field, type the number of times the adapter retries the connection before reporting a polling error. For more information, see “Number of times to retry the system connection (RetryLimit)” on page 141.
 - e. If you want the adapter to stop if polling errors occur, select the **Stop the adapter when an error is encountered while polling** check box. If you do not select this option, the adapter logs an exception but continues to run. For more information, see “Stop the adapter when an error is encountered while polling (StopPollingOnError)” on page 143.
 - f. Select the **Retry EIS connection on startup** check box, if you want the adapter to retry a failed connection when starting. For more information, see “Retry EIS connection on startup (RetryConnectionOnStartup)” on page 143.
 - g. Select the calendar based scheduling option to create calendar based polling for inbound activities. You can schedule your business activities, when you create a new calendar in IBM Integration Designer. The option of working with the calendar based scheduling feature is only possible with IBM Integration Designer as the tooling environment. The following figure helps you to schedule a calendar polling option.

Figure 27. Polling based on calendar

You can either select a blank calendar or create a new calendar for a module or library. When you select a blank calendar, you will not be able to set pre-defined time intervals. You have to define your time intervals. When you create a calendar using a pre-defined template, you can define time intervals for each template.

- 1) Click **New** to create a new calendar entry for a module or library.
 - You can choose an existing calendar, or create a new calendar instance.
 - Click **Browse** to select an existing calendar module. Or click **New** to create a module for the new calendar.
 - Click **Browse** to choose a folder for the calendar. (Optional).
 - Enter a name for the new calendar.
 - Click **Next** if you want to generate the calendar, through a predefined template. Or, click **Finish**, to create a non template calendar.
- 2) Click **Browse** to select an existing calendar for a module or library. In the **Select a Business Calendar** screen you can search for all currently existing calendar files (*.cal) in the IBM Integration Designer workspace.
 - In the **Name** field, type the calendar name or click the calendar in the **Matching business calendars** screen. Click **OK** to open the external service wizard.
 - In the **WebSphere Integration workspace**, select the **Calendar** module, and browse **Integration logic->Calendars**, to view or modify the calendar schedules. You can modify the intervals and exceptions, or add new entries for these elements. For more information, refer to the information at <http://publib.boulder.ibm.com/infocenter/dmndhelp/v7r5mx/index.jsp?topic=/com.ibm.wbpm.wid.bpel.doc/topics/cbuscal.html>.

Note: You must deploy the Business Calendar module to the same IBM Business Process Manager or WebSphere Enterprise Service Bus instance, along with the inbound application. If you do not map these two connections to the same server instance, the inbound application using the business calendar will by default, poll as there is no calendar configured.

- **Event delivery configuration**

- a. In the **Type of delivery** field, select the delivery method. The methods are described in “Delivery type (DeliveryType)” on page 138.

- b. If you want to ensure that events are delivered only once and to only one export, select the **Ensure assured-once event delivery** check box. This option might reduce performance but does not result in duplicate or missing event delivery. For more information, see “Ensure once-only event delivery (AssuredOnceDelivery)” on page 137.
 - c. In the **Event types to process** field, type a colon-separated list of the business objects for which you want the events delivered. Leave this field blank to receive events for all business object types.
 - d. In the **Adapter Instance for event filtering** field, type the connector id for which you want the events delivered. For more information, see “Event groups to filter (AdapterInstanceEventFilter)” on page 136.
 - e. In the **Retry limit for failed events** field, specify the number of times to try to deliver an event after a delivery failure. For more information, see “Retry limit for failed events (FailedEventRetryLimit)” on page 139.
 - f. In the **Number of connections for event delivery** area, specify the minimum and maximum number of connections to use to deliver events. For more information, see “Minimum connections (MinimumConnections)” on page 141 and “Maximum connections (MaximumConnections)” on page 141.
- **Processing methods for illegal XML characters**
 - a. From the **Illegal XML character processing** drop-down list, select one of the options that correspond to your requirement:
 - **Do not validate illegal XML characters**, if you want to work with the default adapter behavior.
 - **Throw exception if contents in the BO contains illegal XML characters**, if you want to proceed after you get an exception message, and log the illegal XML characters in the trace file, at runtime.
 - **Discard all illegal XML characters and related logs**, if you want the adapter to discard the illegal XML characters, and log the characters in the trace file, at runtime.

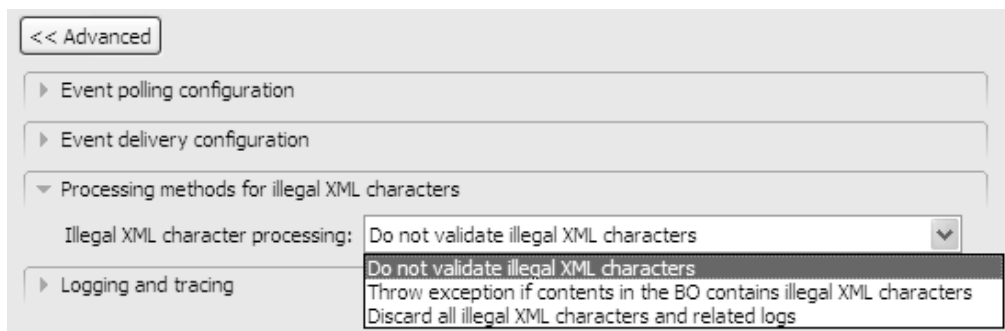


Figure 28. Processing methods for illegal XML characters

- **Logging and tracing**
 - a. If you have multiple instances of the adapter, enter a value in the **Adapter ID** that is unique for this instance. For more information, see “Resource adapter properties” on page 119.
 - b. You can select the **Disguise user data as 'XXX' in log and trace files** check box to prevent sensitive user data from being written to log and trace files.

7. Click **Next**.

Results

The Specify the Location Properties window is displayed.

Related concepts

“Inbound processing” on page 10

WebSphere Adapter for Lotus Domino supports inbound processing of events. The adapter polls the event database in the Lotus Domino server for events at specified intervals. Each time a form is created or updated, the adapter tracks the process as an event. When the adapter detects an event, it requests a copy of the form, converts the form data into a business object, and sends it to the consuming service.

“User authentication” on page 26

The adapter supports several methods for supplying the user name and password that are needed to connect to the Lotus Domino 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 135

Activation specification properties are properties that hold the inbound event processing configuration information for a message endpoint.

“Resource adapter properties” on page 119

The resource adapter properties control the general operation of the adapter, such as specifying the namespace for business objects. You set the resource adapter properties using the external service wizard when you configure the adapter. After deploying the adapter, use the administrative console to change these properties.

“Inbound configuration properties” on page 130

WebSphere Adapter for Lotus Domino 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 IBM Integration Designer or the administrative console, but connection properties for the external service wizard cannot be changed after deployment.

Globalization

WebSphere Adapter for Lotus Domino 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.

Setting location properties for artifacts

After you select and configure deployment properties, use the external service wizard to configure location properties that the adapter uses to generate the service.

Before you begin

This task is performed using the Specify the Location Properties window of the external service wizard.

About this task

To configure location properties for your module follow this procedure:

Procedure

1. In the Specify the Location Properties window, either select an existing module or create a new one.
 - To use an existing module, select its name from the **Module** list.

Important: If the module contains an interface or business object with the same name as that you are now configuring, the original interface or business object in the module is replaced by the new version.

- To create a new module:
 - a. Click **New**.
 - b. In the Select a Business Integration Project Type window, select **Module** and click **Next**.
 - c. In the Create a Module window, type a name for the module in the **Module name** field. An example of a module name is DominoInboundInterface.
 - d. If you want the service description files (the `.import` and the `.wsdl` files) to be located in the default folder in the module, leave the **Use default location** check box selected. If you want to specify a different folder in the module, clear the option and then click **Browse** to specify a different folder in the **Location** field.
 - e. To automatically open the module in the assembly diagram in Integration Designer when the wizard closes, select the **Open the module assembly diagram** check box. Otherwise, clear this option.
 - f. Click **Finish** to create the module.
2. Specify the namespace you want to use for your inbound interface.
 - If you want the inbound interface to use the default derived namespace, leave **Use the default namespace** check box selected.
 - To specify a different namespace, clear the option and type a different value in the **Namespace** field.
 3. Specify the folder within the new module where the service description is to be saved. In the **Folder** field, type the folder name or browse to an existing folder. If you do not specify a folder name, the artifacts (the import, XSD, and WSDL files) are stored in the root folder of the module (the folder with the module name).
 4. In the **Name** field, accept the default import name or type a different name.
 5. If you want to save business objects in a library where they can be used by other modules, select the **Save business objects to a library** check box and specify the location of the library in the **Library** field.
 6. In the **Description** field, type a descriptive comment about the module.
 7. Click **Finish** to complete the setting of location properties.

Note: If the Model Changed window is displayed, click **Yes**.

Results

The wizard exits. The artifacts are generated.

Related concepts

“Inbound processing” on page 10

WebSphere Adapter for Lotus Domino supports inbound processing of events. The adapter polls the event database in the Lotus Domino server for events at specified intervals. Each time a form is created or updated, the adapter tracks the process as an event. When the adapter detects an event, it requests a copy of the form, converts the form data into a business object, and sends it to the consuming service.

Related reference

“Activation specification properties” on page 135

Activation specification properties are properties that hold the inbound event processing configuration information for a message endpoint.

“Resource adapter properties” on page 119

The resource adapter properties control the general operation of the adapter, such as specifying the namespace for business objects. You set the resource adapter properties using the external service wizard when you configure the adapter. After deploying the adapter, use the administrative console to change these properties.

“Inbound configuration properties” on page 130

WebSphere Adapter for Lotus Domino 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 IBM Integration Designer or the administrative console, but connection properties for the external service wizard cannot be changed after deployment.

Globalization

WebSphere Adapter for Lotus Domino 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.

Chapter 5. Modifying artifacts

Often business requirements mandate changes to the backend enterprise information system (EIS) data structures. These changes call for regeneration and reconfiguration of artifacts (import and export) that were previously generated using the external service wizard.

A few business scenarios where the output from one enterprise service discovery flow can be reused in successive flows, are outlined as follows:

- When you want to add a new object to the object set.
- When you want to modify configurations on selected objects like changes to operations, operation names, and service level settings like security, transactions, and reliability.
- When you want to remove a discovered object from the object set.
- When you want to rediscover existing objects in the service to synchronize the service if the objects in the back-end system have been updated.

To modify existing artifacts, you can invoke the wizard in one of the following ways. The external service wizard is initialized with previously configured settings.

- In the assembly editor, select the component you want to modify, right-click, and select **Edit Binding**.
- In the Business Integration view, select the component you want to modify, right-click, and select **Edit Binding**.
- Select the component in the assembly editor and select the Properties view. In the Binding tab, click the **Edit** link.

Note: The Edit Binding option is available for artifacts generated using IBM Integration Designer 7.0 only. If you are importing a project interchange from an earlier version of IBM Integration Designer, the Edit Binding option is not available. If you have made any manual changes to the configuration, running the wizard again will overwrite these changes.

Modifying service import

Modify an import component by rediscovering and reconfiguring the objects using the Edit Binding option in IBM Integration Designer.

About this task

You can invoke the external service wizard to modify the information of a service import interface. The wizard automatically populates the existing information for the selected import interface. You can modify the objects and services, and then regenerate the import component with the modified data.

Procedure



1. Invoke the external service wizard for the selected service interface import component using one of the following methods.
 - In the assembly editor, select the component you want to modify, right-click and select **Edit Binding**.
 - In the Business Integration view, select the component you want to modify, right-click and select **Edit Binding**.

- Select the interface in the assembly editor and select the Properties view. In the Binding tab, click the **Edit** link.

The Find Objects in the Enterprise System window of the external service wizard is displayed. The external service wizard automatically populates the existing configuration details for the selected import interface.

2. In the Find Objects in the Enterprise System window, make the required changes. For more information about discovering objects, see “Discovering and selecting objects” on page 46.

Note: To change the connection properties for the external service wizard, click **Back** and change the properties in the Specify the Discovery Properties window. For more information see, “Setting connection properties for the external service wizard” on page 45.

- a. To modify the configuration of an object from the **Selected objects** list, select the object name and then click the  (Edit) icon.
- b. To remove an object from the **Selected objects** list, select the object name, and then click the  (Remove) button.

3. Click **Next**. If you click **Cancel**, the changes you made in the previous step does not take effect.
4. In the Specify Composite Properties window, you can modify the existing operations or add more operations to the selected object. For more information about the configuration procedure, refer to “Configuring the selected objects” on page 50.
5. Click **Next**.
6. In the Service Generation window, modify the service operations if required.
7. Click **Finish**. The artifacts are regenerated.
8. Complete any other required manual configuration.

Results

The artifacts are updated.

What to do next

You can test and deploy your module.

Modifying service export

Modify an export component by rediscovering and reconfiguring the objects using the Edit Binding option in IBM Integration Designer.

About this task

You can invoke the external service wizard to modify the information of a service export interface. The wizard automatically populates the existing information for the selected export interface. You can modify the objects and services, and then regenerate the export component with the modified data.

Procedure



1. Invoke the external service wizard for the selected service interface export component using one of the following methods.

- In the assembly editor, select the component you want to modify, right-click and select **Edit Binding**.
- In the Business Integration view, select the component you want to modify, right-click and select **Edit Binding**.
- Select the interface in the assembly editor and select the Properties view. In the Binding tab, click the **Edit** link.

The Find Objects in the Enterprise System window of the external service wizard is displayed. The external service wizard automatically populates the existing configuration details for the selected export interface.

2. In the Find Objects in the Enterprise System window, make the required changes. For more information about discovering objects, see “Discovering and selecting objects” on page 58.

Note: To change the connection properties for the external service wizard, click **Back** and change the properties in the Specify the Discovery Properties window. For more information see, “Setting connection properties for the external service wizard” on page 45.

- a. To modify the configuration of an object from the **Selected objects** list, select the object name and then click the  (Edit) icon.
 - b. To remove an object from the **Selected objects** list, select the object name, and then click the  (Remove) button.
3. Click **Next**. If you click **Cancel**, the changes you made in the previous step does not take effect.
 4. In the Specify Composite Properties window, you can modify the existing operations or add more operations to the selected object. For more information about the configuration procedure, refer to “Configuring the selected objects” on page 62.
 5. Click **Next**.
 6. In the Service Generation window, modify the service operations if required.
 7. Click **Finish**. The artifacts are regenerated.
 8. Complete any other required manual configuration.

Results

The artifacts are updated.

What to do next

You can test and deploy your module.

Chapter 6. Changing interaction specification properties

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

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 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:

Procedure

1. From the Business Integration perspective of IBM Integration Designer, expand the module name.
2. Expand **Assembly Diagram** and double-click the interface.
3. Click the interface in the assembly editor. The module properties are displayed.
4. Click the **Properties** tab. You can also right-click the interface in the assembly 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.

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 127

Interaction specification properties control the interaction for an operation. The external service wizard sets the interaction specification properties when you configure the adapter. Typically, you do not need to change these properties. However, you can change some properties for outbound operations. To change these properties after the application is deployed, use the assembly editor in IBM Integration Designer. The properties are located in the method binding of the import.

Chapter 7. 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 IBM Integration Designer, the integrated test environment features runtime support for IBM Business Process Manager 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 IBM Integration Designer, 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 IBM Business Process Manager or WebSphere Enterprise Service Bus as EAR files using the administrative console or command-line tools.

Deploying the module for testing

In IBM Integration Designer, 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.

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 IBM Integration Designer. The adapter uses the wire to pass event data (from the export to the target component).

Procedure

1. Create the target component.

- a. From the Business Integration perspective of IBM Integration Designer, 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.
 - 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 > Java**.
 - 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 IBM Integration Designer, 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, generate and wire a *target component* to which the adapter sends the 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.

Procedure

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 servers settings.
 - d. Click **Finish** to publish the server.

2. Add the module to the server.
 - a. Switch to the servers view. In IBM Integration Designer, select **Windows > Show View > Servers**.
 - a. Start the server. In the **Servers** tab in the lower-right pane of the IBM Integration Designer screen, right-click the server, and then select **Start**.
3. When the server status is *Started*, right-click 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 IBM Integration Designer integration test client.

Before you begin

You need to add the module to the server first.

About this task

Testing a module is 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.

Procedure

1. Select the module you want to test, right-click on it, and select **Test > Test Module**.
2. For information about testing a module using the test client, see the *Testing modules and components* topic in the IBM Integration Designer information center.

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 IBM Business Process Manager or WebSphere Enterprise Service Bus in a production environment is a two-step process. First, you export the module in IBM Integration Designer as an enterprise archive (EAR) file. Second, you deploy the EAR file using the IBM Business Process Manager or WebSphere Enterprise Service Bus administrative console.

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 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 set **Deploy connector project** to **On server for use by multiple adapters** in the Specify the Service Generation and Deployment Properties 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.

Procedure

1. If the server is not running, right-click your server in the **Servers** view and select **Start**.
2. When the server status changes to **Started**, right-click the server and select **Administration > Run administrative console**.
3. Log on to the administrative console.
4. Click **Resources > Resource Adapters > Resource adapters**.
5. In the Resource adapters page, click **Install RAR**.

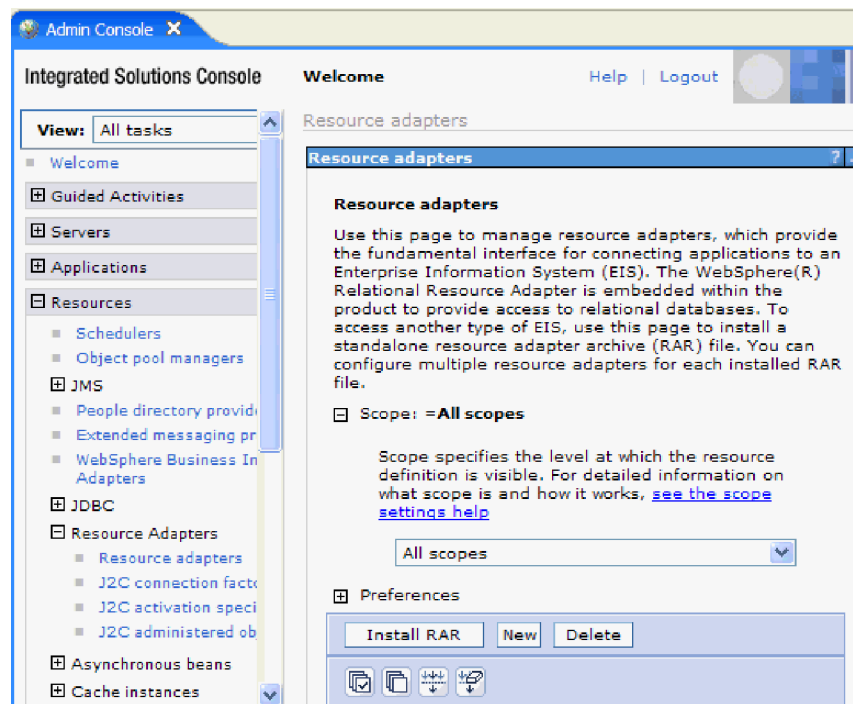


Figure 29. The Install RAR button on the Resource adapters page

6. 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:
IID_installation_directory/ResourceAdapters/adapter_name/adapter.rar
7. Click **Next**.
8. Optional: In the Resource adapters page, change the name of the adapter and add a description.
9. Click **OK**.
10. 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 IBM Integration Designer, 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 IBM Business Process Manager 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 IBM Integration Designer Business Integration perspective.

About this task

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

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. To export the source files, select the **Export source files** check box. 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 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 IBM Business Process Manager 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 IBM Business Process Manager or WebSphere Enterprise Service Bus.

About this task

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

Procedure

1. If the server is not running, right-click your server in the **Servers** view and select **Start**.
2. When the server status changes to **Started**, right-click the server and select **Administration > Run administrative console**.
3. Log on to the administrative console.
4. Click **Applications > New Application > New Enterprise Application**.



Figure 30. Preparing for the application installation window

5. Click **Browse** to locate your EAR file and click **Next**. The EAR file name is the name of the module followed by "App."

6. Optional: If you are deploying to a clustered environment, complete the following steps.
 - a. On the **Step 2: Map modules to servers** window, select the module and click **Next**.
 - b. Select the name of the server cluster.
 - c. Click **Apply**.
7. Click **Next**. In the Summary page, verify the settings and click **Finish**.
8. Optional: If you are using an authentication alias, complete the following steps:
 - a. Expand **Security** and select **Business Integration Security**.
 - b. Select the authentication alias that you want to configure. You must have administrator or operator rights 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 8. 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 the configuration properties after you deploy the adapter as part of a module, you use the administrative console of the runtime environment.

Related reference

“Configuration properties” on page 115

IBM WebSphere Adapter for Lotus Domino 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 IBM Business Process Manager 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 IBM Business Process Manager or WebSphere Enterprise Service Bus.

About this task

Custom properties are default configuration properties shared by all IBM WebSphere Adapters.

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

Procedure

1. If the server is not running, right-click your server in the **Servers** view and select **Start**.
2. When the server status changes to **Started**, right-click the server and select **Administration > Run administrative console**.
3. Log on to the administrative console.
4. Select **Applications > Application Types > WebSphere enterprise application**.
5. From the Enterprise Applications list, click the name of the adapter module whose properties you want to change. The **Configuration** page is displayed.

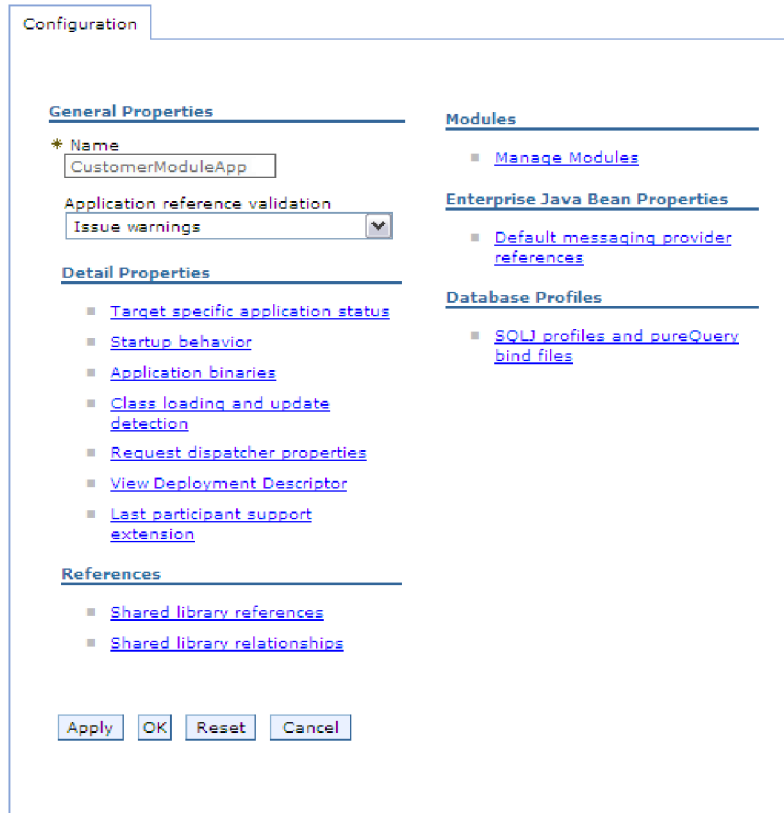


Figure 31. The Manage Modules selection in the Configuration tab

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

Note: See “Resource adapter properties” on page 119 for more information about these properties.

- a. Click the name of the property. The **Configuration** page for the selected property is displayed.
 - b. Change the contents of the **Value** field or type a value, if the field is empty.
 - c. Click **OK**.
11. In the Messages area, click **Save**.

Results

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

Related reference

“Resource adapter properties” on page 119

The resource adapter properties control the general operation of the adapter, such as specifying the namespace for business objects. You set the resource adapter properties using the external service wizard when you configure the adapter. After deploying the adapter, use the administrative console to change these properties.

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 IBM Business Process Manager or WebSphere Enterprise Service Bus.

About this task

You use managed connection factory properties to configure the target Lotus Domino 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.

Procedure

1. If the server is not running, right-click your server in the **Servers** view and select **Start**.
2. When the server status changes to **Started**, right-click the server and select **Administration > Run administrative console**.
3. Log on to the administrative console.
4. Select **Applications > Application Types > WebSphere enterprise application**.
5. In the Enterprise Applications list, click the name of the adapter module whose properties you want to change.

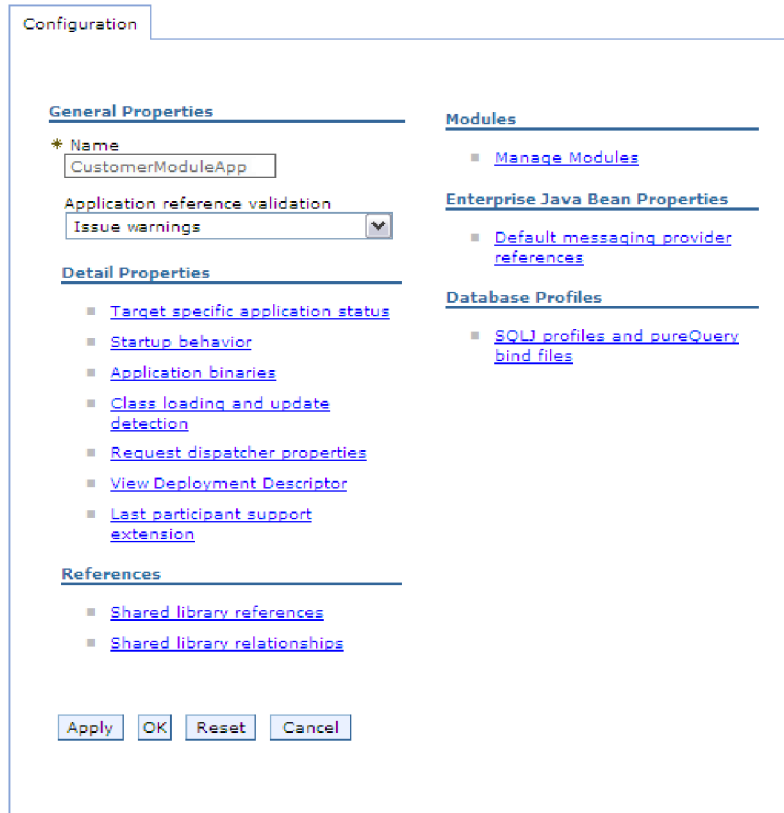


Figure 32. The Manage Modules selection in the Configuration tab

6. Under **Modules**, click **Manage Modules**.
7. Click **IBM WebSphere Adapter for Lotus Domino**.
8. In the **Additional Properties** list, click **Resource Adapter**.
9. On the next page, from the **Additional Properties** list, click **J2C connection factories**.
10. Click the name of the connection factory associated with your adapter module.
11. In the **Additional Properties** list, click **Custom properties**.
 Custom properties are those J2C connection factory properties that are unique to IBM WebSphere Adapter for Lotus Domino. Connection pool and advanced connection factory properties are properties you configure if you are developing your own adapter.
12. For each property you want to change, perform the following steps.

Note: See “Managed (J2C) connection factory properties” on page 121 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**.
13. In the Messages area, click **Save**.

Results

The managed connection factory properties associated with your adapter module are changed.

Related reference

“Managed (J2C) connection factory properties” on page 121

Managed connection factory properties are used by the adapter at run time to create an outbound connection instance with the Lotus Domino 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 IBM Business Process Manager 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.

Procedure

1. If the server is not running, right-click your server in the **Servers** view and select **Start**.
2. When the server status changes to **Started**, right-click the server and select **Administration > Run administrative console**.
3. Log on to the administrative console.
4. Select **Applications > Application Types > WebSphere enterprise application**.
5. From the Enterprise Applications list, click the name of the adapter module whose properties you want to change.

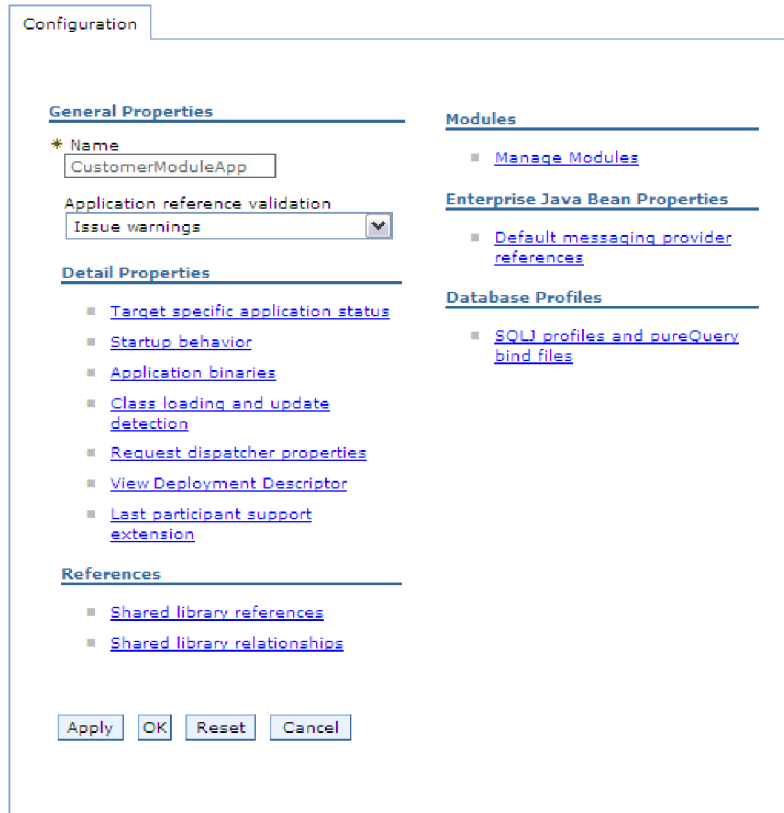


Figure 33. The Manage Modules selection in the Configuration tab

6. Under **Modules**, click **Manage Modules**.
7. Click **IBM WebSphere Adapter for Lotus Domino**.
8. From the **Additional Properties** list, click **Resource Adapter**.
9. On the next page, from the **Additional Properties** list, click **J2C activation specifications**.
10. Click the name of the activation specification associated with the adapter module.
11. From the **Additional Properties** list, click **J2C activation specification custom properties**.
12. For each property you want to change, perform the following steps.
 - 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**.
13. In the Messages area, click **Save**.

Results

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

Related reference

“Activation specification properties” on page 135

Activation specification properties are properties that hold the inbound event processing configuration information for a message endpoint.

Changing configuration properties for stand-alone adapters

To set configuration properties after you install a stand-alone adapter, use the administrative console of the runtime environment. Provide the general information about the adapter and then set the resource adapter properties (which are used for general adapter operation). If the adapter is used for outbound operations, create a connection factory and then set the 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 IBM Business Process Manager 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 IBM Business Process Manager or WebSphere Enterprise Service Bus.

About this task

Custom properties are default configuration properties shared by all IBM WebSphere Adapters.

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

Procedure

1. If the server is not running, right-click your server in the **Servers** view and select **Start**.
2. When the server status changes to **Started**, right-click the server and select **Administration > Run administrative console**.
3. Log on to the administrative console.
4. Click **Resources > Resource Adapters > Resource adapters**.
5. In the Resource adapters page, click **IBM WebSphere Adapter for Lotus Domino**.
6. In the **Additional Properties** list, click **Custom properties**.
7. For each property you want to change, perform the following steps.

Note: See “Resource adapter properties” on page 119 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**.
8. In the Messages area, click **Save**.

Results

The resource adapter properties associated with your adapter are changed.

Related reference

“Resource adapter properties” on page 119

The resource adapter properties control the general operation of the adapter, such as specifying the namespace for business objects. You set the resource adapter properties using the external service wizard when you configure the adapter. After deploying the adapter, use the administrative console to change these properties.

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 IBM Business Process Manager 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 IBM Business Process Manager or WebSphere Enterprise Service Bus.

About this task

You use managed connection factory properties to configure the target Lotus Domino 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:

Procedure

1. If the server is not running, right-click your server in the **Servers** view and select **Start**.
2. When the server status changes to **Started**, right-click the server and select **Administration > Run administrative console**.
3. Log on to the administrative console.
4. Click **Resources > Resource Adapters > Resource adapters**.
5. In the Resource adapters page, click **IBM WebSphere Adapter for Lotus Domino**.
6. In the **Additional Properties** list, click **J2C connection factories**.
7. If you are going to use an existing connection factory, skip ahead to select from the list of existing connection factories.

Note: If you have selected **Specify connection properties** when you use 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 can type AdapterCF.
- c. Type a value for **JNDI name**. For example, you can type com/eis/AdapterCF.
- d. Optional: Select an authentication alias from the **Component-managed authentication alias** list.
- e. Click **OK**.
- f. In the Messages area, click **Save**.

The newly created connection factory is displayed.

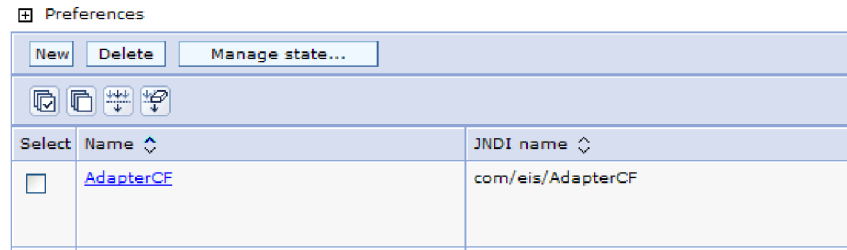


Figure 34. User-defined connection factories for use with the resource adapter

8. In the list of connection factories, click the one you want to use.
9. In the **Additional Properties** list, click **Custom properties**.
Custom properties are those J2C connection factory properties that are unique to WebSphere Adapter for Lotus Domino. 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 (J2C) connection factory properties” on page 121 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. After you have finished setting properties, click **Apply**.
12. In the Messages area, click **Save**.

Results

The managed connection factory properties associated with your adapter are set.

Related reference

“Managed (J2C) connection factory properties” on page 121
Managed connection factory properties are used by the adapter at run time to create an outbound connection instance with the Lotus Domino 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 IBM Business Process Manager 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 IBM Business Process Manager 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.

Procedure

1. If the server is not running, right-click your server in the **Servers** view and select **Start**.
2. When the server status changes to **Started**, right-click the server and select **Administration > Run administrative console**.
3. Log on to the administrative console.
4. Click **Resources > Resource Adapters > Resource adapters**.
5. In the Resource adapters page, click **IBM WebSphere Adapter for Lotus Domino**.
6. In the **Additional Properties** list, click **J2C activation specifications**.
7. If you are going to use an existing activation specification, skip ahead to select from an existing list of activation specifications.

Note: If you have selected **Use predefined connection properties** when you use the external service wizard to configure the adapter module, you must 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 can type AdapterAS.
- c. Type a value for **JNDI name**. For example, you can type com/eis/AdapterAS.
- d. Optional: 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.
8. In the list of activation specifications, click the one you want to use.
9. In the Additional Properties list, click **J2C activation specification custom properties**.
10. For each property you want to set, perform the following steps.
 - 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. After you have finished setting properties, click **Apply**.
12. In the Messages area, click **Save**.

Results

The activation specification properties associated with your adapter are set.

Related reference

“Activation specification properties” on page 135

Activation specification properties are properties that hold the inbound event processing configuration information for a message endpoint.

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.

Procedure

1. If the server is not running, right-click your server in the **Servers** view and select **Start**.
2. When the server status changes to **Started**, right-click the server and select **Administration > Run administrative console**.
3. Log on to the administrative console.
4. Click **Applications > Application Types > WebSphere enterprise applications**.

Note: The administrative console is labeled “Integrated Solutions Console”.

5. Select the application that you want to start. The application name is the name of the EAR file you installed, without the .EAR file extension.
6. 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.

Procedure

1. If the server is not running, right-click your server in the **Servers** view and select **Start**.
2. When the server status changes to **Started**, right-click the server and select **Administration > Run administrative console**.
3. Log on to the administrative console.
4. Click **Applications > Application Types > WebSphere enterprise applications**.

Note: The administrative console is labeled “Integrated Solutions Console”.

5. Select the application that you want to stop. The application name is the name of the EAR file you installed, without the .EAR file extension.
6. Click **Stop**.

Results

The status of the application changes to **Stopped**, and a message stating that the application has stopped is displayed 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 IBM WebSphere Adapter for Lotus Domino. 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 IBM Business Process Manager or WebSphere Enterprise Service Bus.

About this task

Before you 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 IBM Business Process Manager or WebSphere Enterprise Service Bus website:
<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 to gather the 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, see “Enabling tracing with the Common Event Infrastructure” on page 101.

2. Generate at least one outbound request to produce performance data that you can configure.

Procedure

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.

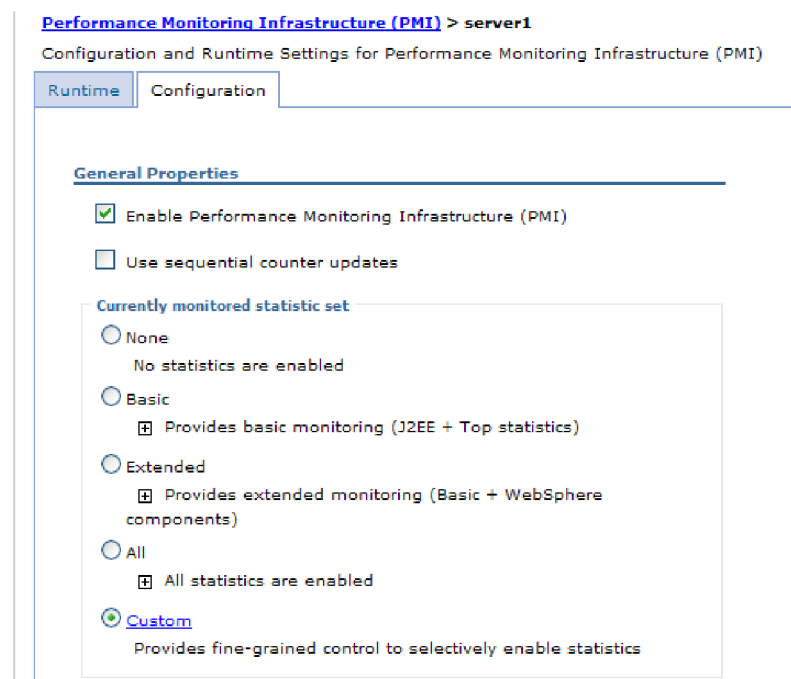


Figure 35. Enabling Performance Monitoring Infrastructure

- 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.

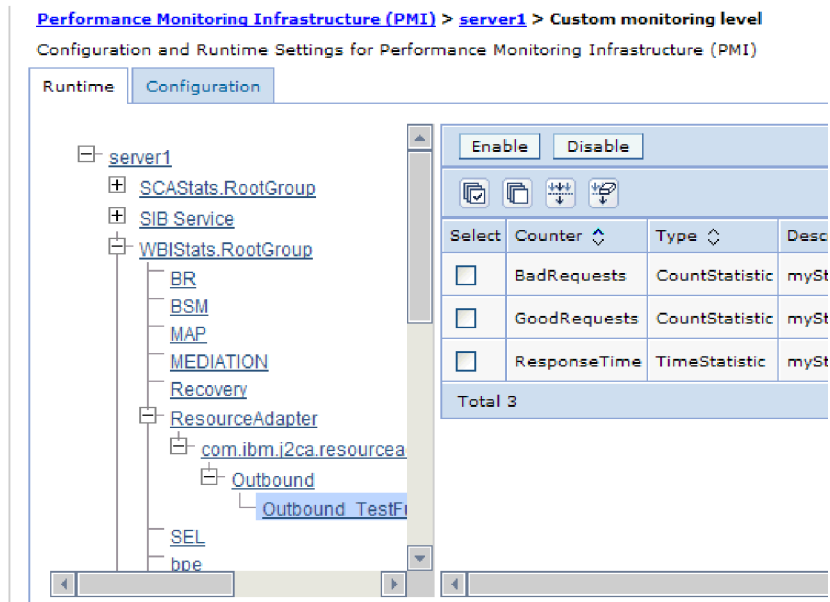


Figure 36. Runtime tab used for configuring PMI

- e. Click **WBISStats.RootGroup**. This is a PMI sub module for data collected in the root group. This example uses the name WBISStats 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 IBM Business Process Manager or WebSphere Enterprise Service Bus.

Before you begin

Configure Performance Monitoring Infrastructure for your adapter.

Procedure

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.

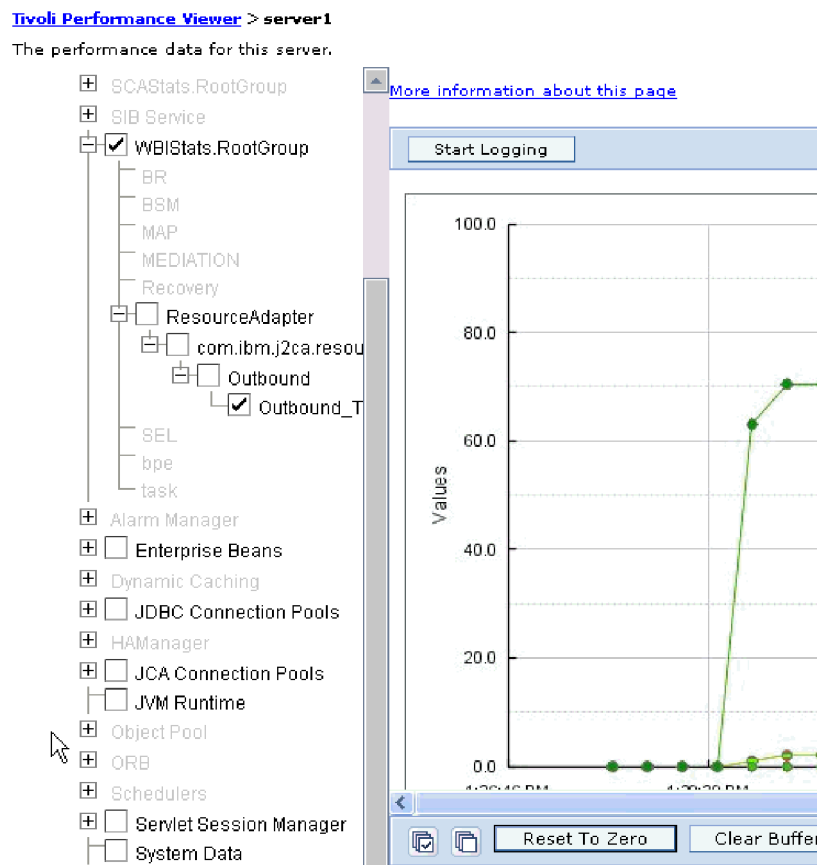


Figure 37. Adapter performance statistics, using graph view

Enabling tracing with the Common Event Infrastructure

The adapter can use the Common Event Infrastructure (CEI), 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

Use this procedure to report CEI entries in the trace log file by using the Common Base Event Browser within the administrative console.

Procedure

1. In the administrative console, click **Troubleshooting**.
2. Click **Logs and Trace**.
3. From 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, select one of the following options:
 - 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 9. Troubleshooting and support

Common troubleshooting techniques and self-help information help you identify and solve problems quickly.

Related reference

“Adapter messages” on page 151

View the messages issued by WebSphere Adapter for Lotus Domino at the following location.

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.

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 IBM 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.

Note: For more information about monitoring on a IBM Process Server, including service components and event points, see http://publib.boulder.ibm.com/infocenter/dmndhelp/v7r5mx/topic/com.ibm.wbpm.admin.doc/topics/welcome_wps_mon.html.

You can change the log configuration statically or dynamically. Static configuration takes effect when you start or restart the application server. Dynamic or run time 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 child logs, which recursively propagate the change to their child log, as necessary.

To enable logging and set the output properties for a log, use the following procedure.

Procedure

1. In the navigation pane of the administrative console, select **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 WebSphere Adapter for Lotus Domino only, select the **com.ibm.j2ca.domino.*** 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 IBM 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 IBM Process Server is written to the `SystemOut.log` and `trace.log` files.

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.

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 Lotus Domino.
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.

Avoiding stale connection problems when invoking the outbound request

Connection-related problems can be resolved in WebSphere Adapter for Lotus Domino using the **connectionRetryLimit** and **connectionRetryInterval** properties defined in the Managed connection factory of the Domino adapter.

Before you begin

The two properties are used to provide two features during outbound communication of the adapter and are optional by default.

About this task

In the service generation and deployment configuration window as explained in the “Setting deployment properties and generating the service” on page 51 topic for the appropriate interface, configure the connection information.

Procedure

1. Click **Advanced** -> **Additional connection configuration**.
2. Set the **Maximum number of retries in case of system connection failure** to the appropriate positive integer:
 - if the property value is set to 0, then the adapter does not perform any EIS connection validation but only executes the outbound operation. If the EIS connection is invalid, the outbound operation fails. Though the subsequent requests get executed successfully provided the Domino system is functional, the current request fails.
 - if the property value is set to greater than 0, then during each request the adapter validates that the EIS connection is active.
 - If connection is valid then operation is completed. If connection is invalid, the adapter invalidates the current managed connection so that a new managed connection is created (new physical connection).
 - If the connection is created successfully, the outbound operation is completed otherwise a `ResourceException` error is thrown.
3. Set the **Time interval between retries if connection fails (milliseconds)** field with the appropriate integer to indicate time in milliseconds in between the attempts. This property is enabled only when the **connectionRetryLimit** property has a value greater than 0.

Note: If the **connectionRetryLimit** property is invalid, the adapter tries to create a new connection. When the adapter is unable to create a connection, the adapter tries to reconnect as per the specified value in the **connectionRetryLimit** property. If the adapter is still unable to create a connection, an exception is thrown.

Results

The two new features take care of connections which are timed out or become stale after the EIS restarts. Even though this option solves most of connection-related problems, the adapter is not guaranteed to work 100% error free for connection problems.

Object no longer exists on server exception

The adapter does not work correctly due to user inactivity during the EMD or the run time process.

Symptoms:

During the EMD process, when you click the Run Query option, or during the run time process, an exception, or a Notes® fault is generated. The following exception message is displayed:

```
"NotesException: Object no longer exists on server"
```

Problem:

The session connecting to the Lotus Domino server times out due to user inactivity for a long time in either the EMD or the run time.

Solution:

To correct this problem, change the idle session timeout setting in the Lotus Domino server and in the EMD configuration.

1. Change the idle session time of the Lotus Domino server, that affects both the EMD and the runtime processes.
 - a. In **IBM Domino Administrator**, click the **Configuration** tab.
 - b. Click to expand the **Server** node in the left pane, and then click the **Current Server Document**.
 - c. Click the **Internet Protocols** tab, and then click the **DIOP** tab.
 - d. In the **Basics** section, set the desired value in the **Idle session timeout** field.
2. Enable the retry on connection failure for the module during EMD, that affects only the runtime processes after deployment.
 - a. During the EMD configuration, open the **Additional configuration** section in the managed connection properties group.
 - b. In the **Maximum number of retries in case of system connection failure** field, specify a value greater than 0. This setting enables the IBM Business Process Manager to retry with new connections to Lotus Domino server, when there is connection problem.

org.omg.CORBA.NO_PERMISSION exception

The adapter does not work correctly when the Internet host name of the Lotus Domino server is not same as the system host name of the server.

Symptoms:

During the EMD process, when you click the Run Query option, an exception message is displayed with the information:

```
"org.omg.CORBA.NO_PERMISSION".
```

Problem:

The Internet host name of the Lotus Domino server is not same as the server host name.

Solution:

To correct this problem, configure the host name of the Lotus Domino server to be same as the server host name.

1. In **IBM Domino Administrator**, click the **Configuration** tab.
2. Click to expand the **Server** node in the left pane, and then click the **Current Server Document**.
3. In the **Basics** section, for **Fully qualified Internet host name** field, set the value as same as the system host name of the server.

Server access denied exception

To make Enterprise Metadata Discovery (EMD) function workable, you must enable HTTP clients to browse Domino databases.

Symptoms:

During the EMD process, when you click the Run Query option, an exception message is displayed with the information:

"NotesException: Server access denied".

Problem:

The Lotus Domino server does not allow HTTP clients to browse databases.

Solution:

To correct this problem, configure the Lotus Domino server to allow HTTP clients to browse databases.

1. In **IBM Domino Administrator**, click the **Configuration** tab.
2. Click to expand the **Server** node in the left pane, and then click the **Current Server Document**.
3. Click the **Internet Protocols** tab. The **HTTP** tab is selected by default.
4. In the **R5 Basics** section, select **Yes** for the **Allow HTTP clients to browse Domino databases** option.

Version conflict exception

The adapter returns a version conflict exception message when installing multiple adapters with different versions.

Adapter returns version conflict exception message**Problem**

When you install multiple adapters with different versions of CWYBS_AdapterFoundation.jar, and if a lower version of the CWYBS_AdapterFoundation.jar is loaded during runtime, the adapter will return the ResourceAdapterInternalException error message, due to a version conflict. For example, when you install Oracle E-Business Suite adapter version 7.0.0.3 and WebSphere Adapter for Lotus Domino version 7.5, the following error message is displayed: IBM WebSphere Adapter for Lotus Domino has loaded file:/C:/IBM/WebSphere/ProcServer7/profiles/ProcSrv01/installedConnectors/CWYOE_OracleEBS.rar/CWYBS_AdapterFoundation.jar with version 7.0.0.3.

However, the base level of this jar required is version 7.5. When you install multiple adapters with different CWYBS_AdapterFoundation.jar versions, the adapter returns the ResourceAdapterInternalException message, due to a version conflict.

Solution

Migrate all adapters to the same version level.

For further assistance, visit http://www-947.ibm.com/support/entry/portal/Overview/Software/WebSphere/WebSphere_Adapters_Family.

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 website

The WebSphere Adapters software support website at http://www-947.ibm.com/support/entry/portal/Overview/Software/WebSphere/WebSphere_Adapters_Family provides links to many resources to help you learn about, use, and troubleshoot WebSphere Adapters, including:

- Flashes (alerts about the product)
- Technical information including the product information center, manuals, IBM Redbooks®, and whitepapers
- Educational offerings
- Technotes

Recommended fixes

A list of recommended fixes you must 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 WebSphere Adapter for Lotus Domino, 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

WebSphere Adapter for Lotus Domino 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 10. Reference

Detailed information about business objects, adapter properties (enterprise service discovery properties, resource adapter properties, managed (J2C) connection factory properties, activation specification properties, and interaction specification properties), messages, and related product information is provided for your reference.

Business object information

A business object is a structure that contains the data required by the adapter to perform an operation. The name of the business object is generated by the wizard in accordance with the naming convention for the adapter.

Business object attributes for operations

Business object attributes define the content of a business object, which is a container for the data specified in the attributes. The structure of the data in the database is defined by the business object. The actual data in the database is contained in the defined business object attributes.

Mapping business object attributes to Domino data types

Table 1 lists the mapping between the business object attributes and the Domino form and field types.

Table 4. Mapping business object attributes to Domino data types

| Serial No. | Domino form field type | Domino Item type in the document | Business object attribute type (SDO) |
|------------|------------------------|----------------------------------|--------------------------------------|
| 1 | Text | ITEM.TEXT | String |
| 2 | Date or Time | ITEM.DATETIMES | Date |
| 3 | Number | ITEM.NUMBERS | Double or Integer |
| 4 | Dialog list | ITEM.TEXT | String |
| 5 | Check box | ITEM.TEXT | String |
| 6 | Radio button | ITEM.TEXT | String |
| 7 | List box | ITEM.TEXT | String |
| 8 | Combination box | ITEM.TEXT | String |
| 9 | Rich Text | ITEM.RICHTEXT | Rich Text business object structure |
| 10 | Authors | ITEM.USERID | String |
| 11 | Names | ITEM.TEXT | String |
| 12 | Readers | ITEM.TEXT | String |
| 13 | Password | ITEM.TEXT | String |
| 14 | Formula | ITEM.FORMULA | String |
| 15 | Time zone | ITEM.TEXT | String |
| 16 | Rich Text Lite | ITEM.RICHTEXT | Rich Text business object structure |
| 17 | Color | ITEM.TEXT | String |

Business object attributes for Rich Text data type

Table 5 lists the properties of a business object attribute for the rich text data type and describes their interpretation and settings.

Table 5. Attribute properties

| Property name | Attribute type | Description |
|---------------|----------------|---|
| Overwrite | Boolean | This property is valid only for the update operation. It determines if the input attachments and texts overwrite or append to the existing rich text data type. |
| Texts | String | The text content of the Rich Text field. |
| Attachments | Attachment | The attachments of the Rich Text field. |

Business object attributes for Rich Text complex item data type

Table 6 lists the properties of a business object attribute for the rich text complex item data type and describes their interpretation and settings.

Table 6. Attribute properties

| Property name | Attribute type | Description |
|---------------|------------------|--|
| Overwrite | Boolean | This property is valid as input only for the update operation. It determines if the input attachments and texts overwrite or append to the existing Rich Text data type. |
| Texts | String array | The text content of the Rich Text complex item field. |
| Attachments | Attachment array | The attachments of the Rich Text complex item field. |
| ValueLength | Integer | The complex items length in bytes. |
| isEncrypted | Boolean | The value is true, if the complex item is encrypted. |
| isProtected | Boolean | The value is true, if editor access is required to modify the complex item. |
| isSigned | Boolean | The value is true, if the complex item is signed. |
| isSummary | Boolean | The value is true, if the complex item value can be shown in a summarized view. |

Business object attributes for Attachment data type

Table 7 lists the properties of a business object attribute for the rich text data type and describes their interpretation and settings.

Table 7. Attribute properties

| Property name | Attribute type | Description |
|----------------------|----------------|---|
| FilePath | String | The absolute file path where the attachment is located. |
| RetrievedContentSize | Integer | The size of the attachment. |
| RetrievedContent | hexBinary | The binary representation of the attachment. |

Business object attributes for complex item data type except for Text complex item

Table 8 lists the properties of a business object attribute for the complex item data type and describes their interpretation and settings.

Table 8. Attribute properties

| Property name | Attribute type | Description |
|---------------|--|---|
| Value | Date, Date array, Double, Double array, Integer, Integer array | The value of the complex item and its data type is determined by the correspondent Lotus Domino type. |
| ValueLength | Integer | The complex items length in bytes. |
| isEncrypted | Boolean | The value is true, if the complex item is encrypted. |
| isProtected | Boolean | The value is true, if editor access is required to modify the complex item. |
| isSigned | Boolean | The value is true, if the complex item is signed. |
| isSummary | Boolean | The value is true, if the complex item value can be shown in a summarized view. |
| LastModified | Date | The last modified date of the complex items. |
| Name | String | The name of the complex item. |
| Type | String | The type of the complex item. |

Business object attributes for Text complex item

Table 9 lists the properties of a business object attribute for the complex item data type and describes their interpretation and settings.

Table 9. Attribute properties

| Property name | Attribute type | Description |
|---------------|------------------------|--|
| Value | String or String array | The value of the complex item, its data type is determined by the corresponding Lotus Domino field type. When the corresponding Lotus Domino field allows multi-value, its data type is array. |
| IsAuthors | Boolean | The value is true, if the values are to be evaluated as Authors. |
| IsNames | Boolean | The value is true, if the value of the text complex item is to be evaluated as Names. |
| IsReaders | Boolean | The value is true, if the value of the text complex item is to be evaluated as Readers. |
| ValueLength | Integer | The length of the complex items in bytes. |
| isEncrypted | Boolean | The value is true, if the complex item is encrypted. |
| isProtected | Boolean | The value is true, if editor access is required to modify the complex item. |
| isSigned | Boolean | The value is true, if the complex item is signed. |
| isSummary | Boolean | The value is true, if the complex item value can be shown in a summarized view. |
| LastModified | Date | The last modified date of the complex items. |

Table 9. Attribute properties (continued)

| Property name | Attribute type | Description |
|---------------|----------------|-------------------------------|
| Name | String | The name of the complex item. |
| Type | String | The type of the complex item. |

Naming conventions

When the external service wizard generates a business object, it provides a name for the business object that reflects the naming convention for the adapter. Typically, the business object name indicates the structure of the business object.

The name of the business object is based on the name of the object in the Lotus Domino server. You use the Business Object Editor to create form-based objects.

The name of a form-based business object is constructed based on the database file path and form name in the Lotus Domino server. For example, when the form Document in database dir1/TestApp1.nsf is selected through the external service wizard, the name of the generated form-based business object is Dir1TestApp1Document.

Note: White spaces in the business object name are removed. Also, if either the database file path or the form name contains unqualified name character, that character is replaced with its corresponding Unicode character. For example, when the form (Archive Log) in the database TestApp1.nsf is selected through the external service wizard, the name of the generated form-based business object is TestApp1U40ArchiveLogU41.

You can rename business objects by using the refactoring functionality in Integration Designer. For more details, refer to the <http://publib.boulder.ibm.com/infocenter/dmndhelp/v7r5mx/topic/com.ibm.wbpm.wid.bpel.doc/selector/topics/trefacts.html> documentation.

Fault business objects

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.

The Domino adapter provides support for outbound fault handling. The wizard creates the following fault business objects:

- DocumentNotFoundFault

The adapter generates this fault when an invalid Universal ID or Note ID is specified during the Update, Delete, or Retrieve operations and the adapter is unable to retrieve the document to complete the operation. This fault can occur for the Update, Delete, and Retrieve operations.

- InvalidRequestFault

The adapter generates this fault when the Universal ID or Note ID is not specified during the Update, Delete, Retrieve, or Exists operations. This fault can occur for the Update, Delete, Retrieve, and Exists operations.

- InvalidParentDocumentFault

The adapter generates this fault when invalid Universal ID is specified in the ParentDocument Universal ID property in the request business object. This fault can occur for the Create, Update, and RetrieveAll operations.

- **MatchesExceededLimitFault**

When processing a RetrieveAll operation, the adapter generates this fault if the number of documents returned for the input query exceeds the value for the MaxRecords property in the interaction specification. You can avoid this fault by increasing the value specified for the MaxRecords property in the interaction specification properties for the RetrieveAll operation.

- **NotesErrorFault**

The adapter generates this fault during an operation to capture any Domino API-related errors in the fault business object. It also provides information about the reason for the error using the NotesErrorID, NotesErrorText, and NotesErrorInternal attributes. This fault can occur for all the operations.

Note: This fault is not enabled by default and is configured using the “Set business faults behavior during run time ” on page 126 property in the managed connection factory properties.

Related concepts

“Business faults” on page 21

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.

“Create operation” on page 4

The Create operation creates a document in the Lotus Domino server with the specified content.

“Update operation” on page 5

The Update operation updates a document in the Lotus Domino server based on the request from the SCA.

“Exists operation” on page 6

The Exists operation checks if a document exists in the Lotus Domino server.

“RetrieveAll operation” on page 6

The RetrieveAll operation is used to retrieve the documents from the database in the Lotus Domino server that matches the values specified in the search fields.

“Retrieve operation” on page 8

The Retrieve operation extracts a document content based on the request from the SCA.

“Delete operation” on page 9

The Delete operation deletes a document in the Lotus Domino server based on the request from the SCA.

Configuration properties

IBM WebSphere Adapter for Lotus Domino 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 IBM Business Process Manager or WebSphere Enterprise Service Bus.

Guide to information about properties

The properties used to configure WebSphere Adapter for Lotus Domino 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 | <p>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.</p> <p>Removing a default value from a required field on the external service wizard <i>will not change that default value</i>. When a required field contains no value at all, the external service wizard processes the field using its assigned default value, and that default value is displayed on the administrative console.</p> <p>Possible values are Yes and No.</p> <p>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,</p> <ul style="list-style-type: none"> • 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 | <p>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.</p> <p>The word None is an acceptable default value, and does not mean that there is no default value.</p> |
| Unit of measure | Specifies how the property is measured, for example in kilobytes or seconds. |
| Property type | <p>Describes the property type. Valid property types include:</p> <ul style="list-style-type: none"> • Boolean • String • Integer |
| Usage | <p>Describes usage conditions or restrictions that might apply to the property. For instance, here is how a restriction would be documented:</p> <p>For Rational® Application Developer for WebSphere Software version 6.40 or earlier, the password:</p> <ul style="list-style-type: none"> • Must be uppercase • Must be 8 characters in length <p>For versions of Rational Application Developer for WebSphere Software later than 6.40, the password:</p> <ul style="list-style-type: none"> • Is not case sensitive • Can be up to 40 characters in length. <p>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.</p> |

| Row | Explanation |
|----------------|---|
| Example | Provides sample property values, for example: "If Language is set to JA (Japanese), code page 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 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. Valid values are Yes and No . |

Outbound configuration properties

IBM WebSphere Adapter for Lotus Domino 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 IBM Business Process Manager or WebSphere Enterprise Service Bus using IBM Integration Designer or the administrative console, but connection properties for the external service wizard cannot be changed after deployment.

Connection properties for the wizard

External service connection properties are used to establish a connection between the external service wizard, a tool that is used to create business objects, and the Lotus Domino server. These properties specify such things as connection configuration, bidirectional transformation properties, and logging options for the wizard. Once a connection is established, the wizard can discover in the Lotus Domino server the metadata it needs to create business objects.

The following table lists and describes the connection properties. These properties can only be configured using the wizard and cannot be changed after deployment. A complete 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" on page 116.

Table 10. Connection properties for the external service wizard

| Property name in the wizard | Description |
|------------------------------------|---|
| "Host name" | The fully qualified host name or IP address of the host where the Lotus Domino server is running. |
| "DIIOP port number" on page 118 | The port of the Lotus Domino server to be used. |
| "User name (UserName)" on page 118 | The Lotus Domino server user name. |
| "Password (Password)" on page 118 | The password for the corresponding user name. |

Host name

This property specifies the host name or IP address of the machine on which the Lotus Domino server is running.

Table 11. Host name of the Lotus Domino server

| | |
|----------------|-----------------------------|
| Required | Yes |
| Default | No default value |
| Property type | String |
| Usage | Must be a valid server name |
| Example | user.us.ibm.com |
| Globalized | No |
| Bidi supported | No |

DIIOP port number

This property specifies the DIIOP port to be used for accessing data in the Lotus Domino server.

Table 12. DIIOP port number to use on the Lotus Domino server

| | |
|----------------|---|
| Required | Yes |
| Default | 63148 |
| Property type | Integer |
| Usage | Must be a valid Domino IIOp port that allows Java applets and applications to access Domino data remotely using the CORBA standard. |
| Globalized | No |
| Bidi supported | No |

User name (UserName)

This property specifies the user name for connecting to the Lotus Domino server.

Table 13. User name details

| | |
|----------------|---|
| Required | Yes |
| Default | No default value |
| Property type | String |
| Usage | The user name for connecting to the Lotus Domino server. Typing an incorrect user name prevents the adapter from accessing the Lotus Domino server. |
| Globalized | No |
| Bidi supported | No |

Password (Password)

This property specifies the password for the Lotus Domino server user name.

Table 14. Password details

| | |
|---------------|---|
| Required | Yes |
| Default | No default value |
| Property type | String |
| Usage | The password for the Lotus Domino server user name. Typing an incorrect password prevents the adapter from accessing the Lotus Domino server. |

Table 14. Password details (continued)

| | |
|----------------|----|
| Globalized | No |
| Bidi supported | No |

Related concepts

“Outbound processing” on page 3

WebSphere Adapter for Lotus Domino supports outbound request processing. When the adapter receives a request, which is sent in the form of a business object from the module, it processes the request and returns the result as a business object. The adapter provides a standard set of outbound operations.

Related tasks

“Setting deployment properties for a runtime environment” on page 52

The connection properties are initialized to the values that the wizard used to connect to the database. To configure the module to use other values, change values such as user name and password here.

“Setting location properties for artifacts” on page 56

After you select and configure deployment properties, use the external service wizard to configure location properties that the adapter uses to generate the service.

Resource adapter properties

The resource adapter properties control the general operation of the adapter, such as specifying the namespace for business objects. You set the resource adapter properties using the external service wizard when you configure the adapter. After deploying the adapter, use the administrative console to change these properties.

The following table lists the resource adapter properties and their purpose. A complete description of each property is provided in the sections that follow the table.

Table 15. Resource adapter properties for the WebSphere Adapter for Lotus Domino

| Name | | Description |
|--|-------------------------------|---|
| In the wizard | In the administrative console | |
| “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” on page 120 | hideConfidentialTrace | Specifies whether to disguise potentially sensitive information by writing X strings instead of user data in the log and trace files. |

Adapter ID

This property identifies a specific deployment, or instance, of the adapter.

Table 16. Adapter ID details

| | |
|---------------|--------|
| Required | Yes |
| Default | 001 |
| Property type | String |

Table 16. Adapter ID details (continued)

| | |
|----------------|---|
| Usage | <p>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, DMNORA, 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 DMNORA001.</p> <p>If you run multiple instances of the same adapter, ensure that the first characters of the adapter ID property are unique for each instance so that you can correlate the 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 the log and trace information to a particular instance of an adapter.</p> <p>For example, when you set the adapter ID property of two instances of WebSphere Adapter for Lotus Domino to 001 and 002. The component IDs for those instances, DMNORA001 and DMNORA002, 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. If you set the adapter ID properties of two instances to Instance01 and Instance02, you cannot examine the log and trace information for each adapter instance because the component ID for both instances is truncated.</p> <p>For inbound processing, the value of this property is set at the resource adapter level. For outbound processing, the value can be set both at the resource adapter level and the managed connection factory level. After you use 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 IBM Integration Designer assembly editor or the administrative console to reset these properties, ensure that you set them consistently, to prevent inconsistent marking of the log and trace entries.</p> |
| Globalized | Yes |
| Bidi supported | No |

Disguise user data as "XXX" in log and trace files

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 17. Disguise user data as "XXX" in log and trace files details

| | |
|-----------------|--|
| Required | No |
| Possible values | True False |
| Default | False |
| Property type | Boolean |
| Usage | <p>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.</p> <p>For inbound processing, the value of this property is set at the resource adapter level. For outbound processing, the value can be set both at the resource adapter level and the managed connection factory level. After you use 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 IBM Integration Designer assembly editor or the administrative console to reset these properties, ensure that you set them consistently, to prevent inconsistent marking of the log and trace entries.</p> |
| Globalized | No |

Table 17. Disguise user data as "XXX" in log and trace files details (continued)

| | |
|----------------|----|
| Bidi supported | No |
|----------------|----|

Related concepts

"Inbound processing" on page 10

WebSphere Adapter for Lotus Domino supports inbound processing of events. The adapter polls the event database in the Lotus Domino server for events at specified intervals. Each time a form is created or updated, the adapter tracks the process as an event. When the adapter detects an event, it requests a copy of the form, converts the form data into a business object, and sends it to the consuming service.

"Outbound processing" on page 3

WebSphere Adapter for Lotus Domino supports outbound request processing. When the adapter receives a request, which is sent in the form of a business object from the module, it processes the request and returns the result as a business object. The adapter provides a standard set of outbound operations.

Related tasks

"Setting deployment properties for a runtime environment" on page 63

The connection properties are initialized to the values that the wizard used to connect to the database. To configure the module to use other values, change values such as user name and password here.

"Setting location properties for artifacts" on page 69

After you select and configure deployment properties, use the external service wizard to configure location properties that the adapter uses to generate the service.

"Setting deployment properties for a runtime environment" on page 52

The connection properties are initialized to the values that the wizard used to connect to the database. To configure the module to use other values, change values such as user name and password here.

"Setting location properties for artifacts" on page 56

After you select and configure deployment properties, use the external service wizard to configure location properties that the adapter uses to generate the service.

Managed (J2C) connection factory properties

Managed connection factory properties are used by the adapter at run time to create an outbound connection instance with the Lotus Domino server.

You set the managed connection factory properties using the external service wizard and can change them using IBM Integration Designer Assembly Editor, or after deployment through the IBM Business Process Manager administrative console.

The following table lists the managed connection factory properties. A complete 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 understanding property details.

Note: The external service wizard refers to these properties as managed connection factory properties and the IBM Business Process Manager administrative console refers to them as (J2C) connection factory properties.

Table 18. Managed connection factory properties

| Property name | | Description |
|---|--------------------------------------|---|
| In the wizard | In the administrative console | |
| “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” on page 123 | hideConfidentialTrace | Specifies whether to disguise potentially sensitive information by writing X strings instead of user data in the log and trace files. |
| “Host name ” on page 124 | hostName | Specifies the name or address of the server that hosts Domino. |
| “Maximum retries on connection failure” on page 124 | connectionRetryLimit | Specifies the number of times the adapter tries to connect to the Lotus Domino server to reestablish the connection. |
| “Password” on page 125 | password | Specifies the password of the user account of the adapter in the Lotus Domino server. |
| “DIIOP port number” on page 125 | portNumber | Specifies the port number, the adapter uses to access the Lotus Domino server. |
| “Retry interval if connection fails (in milliseconds) entities” on page 125 | connectionRetryInterval | Specifies the time interval between the attempts to reconnect to the Lotus Domino server if the connection fails. |
| “Set business faults behavior during run time ” on page 126 | handleAllFaultExceptions | Specifies the fault handling behavior during run time. |
| “User name ” on page 126 | userName | Specifies the name of the user account of the adapter in the Lotus Domino server. |

Adapter ID

This property identifies a specific deployment, or instance, of the adapter.

Table 19. Adapter ID details

| | |
|---------------|--------|
| Required | Yes |
| Default | 001 |
| Property type | String |

Table 19. Adapter ID details (continued)

| | |
|----------------|---|
| Usage | <p>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, DMNORA, 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 DMNORA001.</p> <p>If you run multiple instances of the same adapter, ensure that the first characters of the adapter ID property are unique for each instance so that you can correlate the 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 the log and trace information to a particular instance of an adapter.</p> <p>For example, when you set the adapter ID property of two instances of WebSphere Adapter for Lotus Domino to 001 and 002. The component IDs for those instances, DMNORA001 and DMNORA002, 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. If you set the adapter ID properties of two instances to Instance01 and Instance02, you cannot examine the log and trace information for each adapter instance because the component ID for both instances is truncated.</p> <p>For inbound processing, the value of this property is set at the resource adapter level. For outbound processing, the value can be set both at the resource adapter level and the managed connection factory level. After you use 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 IBM Integration Designer assembly editor or the administrative console to reset these properties, ensure that you set them consistently, to prevent inconsistent marking of the log and trace entries.</p> |
| Globalized | Yes |
| Bidi supported | No |

Disguise user data as "XXX" in log and trace files

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 20. Disguise user data as "XXX" in log and trace files details

| | |
|-----------------|--|
| Required | No |
| Possible values | True False |
| Default | False |
| Property type | Boolean |
| Usage | <p>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.</p> <p>For inbound processing, the value of this property is set at the resource adapter level. For outbound processing, the value can be set both at the resource adapter level and the managed connection factory level. After you use 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 IBM Integration Designer assembly editor or the administrative console to reset these properties, ensure that you set them consistently, to prevent inconsistent marking of the log and trace entries.</p> |
| Globalized | No |
| Bidi supported | No |

Host name

This property specifies the name or address of the server that hosts Domino.

Table 21. Host name details

| | |
|----------------|---|
| Required | Yes |
| Default | No default value |
| Property type | String |
| Usage | Identifies the server, either by name or IP address, that hosts Domino. |
| Example | 9.26.159.264 |
| Globalized | No |
| Bidi supported | No |

Maximum retries on connection failure

This property specifies the number of times the adapter attempts to reestablish a connection to the Lotus Domino server, when the adapter encounters an error related to the outbound connection.

Table 22. Maximum retries on connection failure property characteristics

| | |
|-----------------|--|
| Required | No |
| Possible values | Integers equal to and greater than zero |
| Default | 0 |
| Property type | Integer |
| Usage | <p>When this property is set to:</p> <p>0</p> <ul style="list-style-type: none">• The adapter does not attempt to reconnect to the Lotus Domino server, if an error occurs while establishing a connection.• The adapter does not verify if the connection to the Lotus Domino server is valid when there is an outbound request during run time. <p>>0</p> <ul style="list-style-type: none">• The adapter attempts to reconnect to the Lotus Domino server for the specified number of times, if an error occurs while establishing a connection.• The adapter verifies if the connection to the Lotus Domino server is valid when there is an outbound request during run time. If the connection is not valid, it is terminated and a new connection is created to process the request. <p>If the adapter fails to establish a connection after trying for the specified number of times, a connection error is generated.</p> <p>If the adapter is successful in reestablishing the connection, the outbound operation is completed.</p> |
| Globalized | No |
| Bidi supported | No |

Password

This property specifies the password of the user account of the adapter on the Lotus Domino server.

Table 23. Password details

| | |
|----------------|---|
| Required | No |
| Default | No default value |
| Property type | String |
| Usage | The restrictions (case, length, and character) are determined by the version of Domino. |
| Globalized | No |
| Bidi supported | Yes |

DIOP port number

This property specifies the port number the adapter uses to access the Lotus Domino server.

Table 24. Port number details

| | |
|----------------|--|
| Required | Yes |
| Default | 63148 |
| Property type | Integer |
| Usage | The port number is required when you use the external service wizard to discover objects and services. |
| Example | 9000 |
| Globalized | No |
| Bidi supported | No |

Retry interval if connection fails (in milliseconds) entities

This property specifies the time interval between attempts to reconnect to the Lotus Domino server if the connection fails.

Table 25. Retry interval if connection fails (milliseconds) property characteristics

| | |
|-----------------|---|
| Required | No |
| Possible values | Integers equal to and greater than 0 |
| Default | 60000 |
| Unit of measure | Milliseconds |
| Property type | Integer |
| Usage | This property is applicable only if the value of the property "Maximum retries on connection failure" on page 124 is set to greater than 0. When the adapter encounters an error while establishing a connection to the Lotus Domino server, this property specifies the time interval the adapter waits between attempts to reestablish a connection. |
| Globalized | No |
| Bidi supported | No |

Set business faults behavior during run time

This property is used to configure the fault handling behavior during run time.

Table 26. Set business faults behavior during run time

| | |
|----------------|--|
| Required | Yes |
| Default | Only business faults supported |
| Property type | String |
| Usage | Sets the business fault behavior during run time. You can configure the fault behavior during run time by using one of the following settings: <ul style="list-style-type: none">• Set to <code>FAULTS_SUPPORTED_INCLUDING_NOTES_ERROR</code> or 2 to handle both the business faults and the Lotus Domino errors as faults.• Set to <code>FAULTS_SUPPORTED</code> or 1 to handle only the business faults.• Set to <code>FAULTS_NOT_SUPPORTED</code> or 0 to disable fault handling support and log the errors as exceptions. |
| Globalized | No |
| Bidi supported | No |

User name

This property specifies the name of the user account, the adapter uses on the Lotus Domino server.

Table 27. User name details

| | |
|----------------|--|
| Required | No |
| Default | No default value |
| Property type | String |
| Usage | The restrictions (case, length, and character) are determined by the Domino version. |
| Example | DV1 |
| Globalized | No |
| Bidi supported | Yes |

Related concepts

“Outbound processing” on page 3

WebSphere Adapter for Lotus Domino supports outbound request processing. When the adapter receives a request, which is sent in the form of a business object from the module, it processes the request and returns the result as a business object. The adapter provides a standard set of outbound operations.

“WebSphere Application Server environment variables” on page 22

WebSphere Application Server environment variables can be used in the external service wizard to specify directory values. You can now change any string property values in inbound and outbound configuration by only changing the environment variables.

Related tasks

“Setting deployment properties for a runtime environment” on page 52

The connection properties are initialized to the values that the wizard used to connect to the database. To configure the module to use other values, change values such as user name and password here.

“Setting location properties for artifacts” on page 56

After you select and configure deployment properties, use the external service wizard to configure location properties that the adapter uses to generate the service.

“Defining WebSphere Application Server environment variables” on page 41

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

Interaction specification properties

Interaction specification properties control the interaction for an operation. The external service wizard sets the interaction specification properties when you configure the adapter. Typically, you do not need to change these properties. However, you can change some properties for outbound operations. To change these properties after the application is deployed, use the assembly editor in IBM Integration Designer. The properties are located in the method binding of the import.

The following table lists the Interaction specification properties. More information about each property is provided in tables in the sections that follow. For information about how to read the property detail tables in the sections that follow, see “Guide to information about properties” on page 116.

Table 28. Interaction specification property for WebSphere Adapter for Lotus Domino

| Property name | Description |
|-----------------------------------|--|
| MaxRecords | Specifies the maximum number of records to be returned during a RetrieveAll operation. |
| “Use indexed search ” on page 128 | Specifies the indexed search method to be used during text searching. |
| ForceSave | Specifies if the document is to be saved during a conflict of Update operation. |
| MakeResponse | Specifies if the document is to be made available as the response of the saved document even if another user modifies and saves the document during an Update operation. |
| MarkAsRead | Specifies if the document is to be marked as read during an Update operation. |

Table 28. Interaction specification property for WebSphere Adapter for Lotus Domino (continued)

| Property name | Description |
|---------------|---|
| ForceDelete | Specifies if the document is to be deleted during a conflict of Delete operation. |

Maximum records for RetrieveAll operation

This property specifies the maximum number of records to retrieve during a RetrieveAll outbound operation.

Table 29. MaxRecords property characteristics

| | |
|----------------|---|
| Required | Yes |
| Default | 100 |
| Attribute type | Integer |
| Usage | If the number of matches in the database exceeds the value of the MaxRecords property, the adapter returns an error. The adapter uses this property to help avoid out-of-memory issues. |
| Globalized | No |
| Bidi supported | No |

Use indexed search

This property specifies if the index-based search is to be used for faster text searching.

Table 30. Use indexed search

| | |
|-----------------|--|
| Required | No |
| Possible values | True False |
| Default | False |
| Property type | Boolean |
| Usage | If you set this property to true, the adapter uses the indexed search method for searching Lotus Domino databases. The adapter searches the text indexed database and returns exact matches of the search string followed by the closer matches. |
| Globalized | No |
| Bidi supported | No |

Force save the document during a conflict

This property specifies if the document is to be force saved during a conflict of Update operation.

Table 31. ForceSave property characteristics

| | |
|----------------|---------|
| Required | No |
| Default | False |
| Attribute type | Boolean |
| Globalized | No |
| Bidi supported | No |

Save the current document as the response of the original document during a conflict

This property specifies if the document is to be made available as the response of the saved document even if another user modifies and saves the document.

Table 32. *MakeResponse* property characteristics

| | |
|----------------|---------|
| Required | No |
| Default | False |
| Attribute type | Boolean |
| Globalized | No |
| Bidi supported | No |

Mark the document as read.

This property specifies if the document is to be marked as read during an Update operation.

Table 33. *MarkAsRead* property characteristics

| | |
|----------------|---------|
| Required | No |
| Default | True |
| Attribute type | Boolean |
| Globalized | No |
| Bidi supported | No |

Force delete the document during a conflict

This property specifies if the document is to be deleted during a conflict of Delete operation.

Table 34. *ForceDelete* property characteristics

| | |
|----------------|---------|
| Required | No |
| Default | False |
| Attribute type | Boolean |
| Globalized | No |
| Bidi supported | No |

Related concepts

“Outbound processing” on page 3

WebSphere Adapter for Lotus Domino supports outbound request processing. When the adapter receives a request, which is sent in the form of a business object from the module, it processes the request and returns the result as a business object. The adapter provides a standard set of outbound operations.

“Business object structures” on page 17

A business object is a structure that consists of data and additional instructions, if any, for processing the data. The adapter supports the form-based business object structure. A form-based type is a content-specific business object that supports specific business object structures (such as, create and retrieve business objects).

“RetrieveAll operation” on page 6

The RetrieveAll operation is used to retrieve the documents from the database in the Lotus Domino server that matches the values specified in the search fields.

Related tasks

“Setting deployment properties for a runtime environment” on page 52

The connection properties are initialized to the values that the wizard used to connect to the database. To configure the module to use other values, change values such as user name and password here.

“Setting location properties for artifacts” on page 56

After you select and configure deployment properties, use the external service wizard to configure location properties that the adapter uses to generate the service.

“Setting deployment properties and generating the service” on page 51

After you select and configure business objects for your module, use the external service wizard to configure properties that the adapter uses to connect to a Lotus Domino server system. The wizard creates a new business integration module where all the artifacts and property values are saved. This task is performed using the Specify the Service Generation and Deployment Properties and Specify the Location Properties windows of the external service wizard.

Chapter 6, “Changing interaction specification properties,” on page 77

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

Inbound configuration properties

WebSphere Adapter for Lotus Domino 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 IBM Integration Designer or the administrative console, but connection properties for the external service wizard cannot be changed after deployment.

Related concepts

“Inbound processing” on page 10

WebSphere Adapter for Lotus Domino supports inbound processing of events. The adapter polls the event database in the Lotus Domino server for events at specified intervals. Each time a form is created or updated, the adapter tracks the process as an event. When the adapter detects an event, it requests a copy of the form, converts the form data into a business object, and sends it to the consuming service.

Related tasks

“Setting deployment properties for a runtime environment” on page 63

The connection properties are initialized to the values that the wizard used to connect to the database. To configure the module to use other values, change values such as user name and password here.

“Setting location properties for artifacts” on page 69

After you select and configure deployment properties, use the external service wizard to configure location properties that the adapter uses to generate the service.

Connection properties for the wizard

External service connection properties are used to establish a connection between the external service wizard, a tool that is used to create business objects, and the Lotus Domino server. These properties specify such things as connection configuration, bidirectional transformation properties, and logging options for the wizard. Once a connection is established, the wizard can discover in the Lotus Domino server the metadata it needs to create business objects.

The following table lists and describes the connection properties. These properties can only be configured using the wizard and cannot be changed after deployment. A complete 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” on page 116.

Table 35. Connection properties for the external service wizard

| Property name in the wizard | Description |
|------------------------------------|---|
| “Host name” on page 117 | The fully qualified host name or IP address of the host where the Lotus Domino server is running. |
| “DIIOP port number” on page 118 | The port of the Lotus Domino server to be used. |
| “User name (UserName)” on page 118 | The Lotus Domino server user name. |
| “Password (Password)” on page 118 | The password for the corresponding user name. |

Host name

This property specifies the host name or IP address of the machine on which the Lotus Domino server is running.

Table 36. Host name of the Lotus Domino server

| | |
|---------------|-----------------------------|
| Required | Yes |
| Default | No default value |
| Property type | String |
| Usage | Must be a valid server name |
| Example | user.us.ibm.com |

Table 36. Host name of the Lotus Domino server (continued)

| | |
|----------------|----|
| Globalized | No |
| Bidi supported | No |

IIOP port number

This property specifies the IIOP port to be used for accessing data in the Lotus Domino server.

Table 37. IIOP port number to use on the Lotus Domino server

| | |
|----------------|---|
| Required | Yes |
| Default | 63148 |
| Property type | Integer |
| Usage | Must be a valid Domino IIOP port that allows Java applets and applications to access Domino data remotely using the CORBA standard. |
| Globalized | No |
| Bidi supported | No |

User name (UserName)

This property specifies the user name for connecting to the Lotus Domino server.

Table 38. User name details

| | |
|----------------|---|
| Required | Yes |
| Default | No default value |
| Property type | String |
| Usage | The user name for connecting to the Lotus Domino server. Typing an incorrect user name prevents the adapter from accessing the Lotus Domino server. |
| Globalized | No |
| Bidi supported | No |

Password (Password)

This property specifies the password for the Lotus Domino server user name.

Table 39. Password details

| | |
|----------------|---|
| Required | Yes |
| Default | No default value |
| Property type | String |
| Usage | The password for the Lotus Domino server user name. Typing an incorrect password prevents the adapter from accessing the Lotus Domino server. |
| Globalized | No |
| Bidi supported | No |

Related concepts

“Outbound processing” on page 3

WebSphere Adapter for Lotus Domino supports outbound request processing. When the adapter receives a request, which is sent in the form of a business object from the module, it processes the request and returns the result as a business object. The adapter provides a standard set of outbound operations.

Related tasks

“Setting deployment properties for a runtime environment” on page 52

The connection properties are initialized to the values that the wizard used to connect to the database. To configure the module to use other values, change values such as user name and password here.

“Setting location properties for artifacts” on page 56

After you select and configure deployment properties, use the external service wizard to configure location properties that the adapter uses to generate the service.

Resource adapter properties

The resource adapter properties control the general operation of the adapter, such as specifying the namespace for business objects. You set the resource adapter properties using the external service wizard when you configure the adapter. After deploying the adapter, use the administrative console to change these properties.

The following table lists the resource adapter properties and their purpose. A complete description of each property is provided in the sections that follow the table.

Table 40. Resource adapter properties for the WebSphere Adapter for Lotus Domino

| Name | | Description |
|--|-------------------------------|---|
| In the wizard | In the administrative console | |
| “Adapter ID” on page 119 | adapterID | Identifies the adapter instance for PMI events and for logging and tracing. |
| “Disguise user data as “XXX” in log and trace files” on page 120 | hideConfidentialTrace | Specifies whether to disguise potentially sensitive information by writing X strings instead of user data in the log and trace files. |

Adapter ID

This property identifies a specific deployment, or instance, of the adapter.

Table 41. Adapter ID details

| | |
|---------------|--------|
| Required | Yes |
| Default | 001 |
| Property type | String |

Table 41. Adapter ID details (continued)

| | |
|----------------|---|
| Usage | <p>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, DMNORA, 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 DMNORA001.</p> <p>If you run multiple instances of the same adapter, ensure that the first characters of the adapter ID property are unique for each instance so that you can correlate the 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 the log and trace information to a particular instance of an adapter.</p> <p>For example, when you set the adapter ID property of two instances of WebSphere Adapter for Lotus Domino to 001 and 002. The component IDs for those instances, DMNORA001 and DMNORA002, 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. If you set the adapter ID properties of two instances to Instance01 and Instance02, you cannot examine the log and trace information for each adapter instance because the component ID for both instances is truncated.</p> <p>For inbound processing, the value of this property is set at the resource adapter level. For outbound processing, the value can be set both at the resource adapter level and the managed connection factory level. After you use 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 IBM Integration Designer assembly editor or the administrative console to reset these properties, ensure that you set them consistently, to prevent inconsistent marking of the log and trace entries.</p> |
| Globalized | Yes |
| Bidi supported | No |

Disguise user data as "XXX" in log and trace files

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 42. Disguise user data as "XXX" in log and trace files details

| | |
|-----------------|--|
| Required | No |
| Possible values | True False |
| Default | False |
| Property type | Boolean |
| Usage | <p>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.</p> <p>For inbound processing, the value of this property is set at the resource adapter level. For outbound processing, the value can be set both at the resource adapter level and the managed connection factory level. After you use 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 IBM Integration Designer assembly editor or the administrative console to reset these properties, ensure that you set them consistently, to prevent inconsistent marking of the log and trace entries.</p> |
| Globalized | No |

Table 42. Disguise user data as "XXX" in log and trace files details (continued)

| | |
|----------------|----|
| Bidi supported | No |
|----------------|----|

Related concepts

"Inbound processing" on page 10

WebSphere Adapter for Lotus Domino supports inbound processing of events. The adapter polls the event database in the Lotus Domino server for events at specified intervals. Each time a form is created or updated, the adapter tracks the process as an event. When the adapter detects an event, it requests a copy of the form, converts the form data into a business object, and sends it to the consuming service.

"Outbound processing" on page 3

WebSphere Adapter for Lotus Domino supports outbound request processing. When the adapter receives a request, which is sent in the form of a business object from the module, it processes the request and returns the result as a business object. The adapter provides a standard set of outbound operations.

Related tasks

"Setting deployment properties for a runtime environment" on page 63

The connection properties are initialized to the values that the wizard used to connect to the database. To configure the module to use other values, change values such as user name and password here.

"Setting location properties for artifacts" on page 69

After you select and configure deployment properties, use the external service wizard to configure location properties that the adapter uses to generate the service.

"Setting deployment properties for a runtime environment" on page 52

The connection properties are initialized to the values that the wizard used to connect to the database. To configure the module to use other values, change values such as user name and password here.

"Setting location properties for artifacts" on page 56

After you select and configure deployment properties, use the external service wizard to configure location properties that the adapter uses to generate the service.

Activation specification properties

Activation specification properties are properties that hold the inbound event processing configuration information for a message endpoint.

Activation specification properties are used during endpoint activation to notify the adapter of eligible event listeners. During inbound processing, the adapter uses these event listeners to receive events before forwarding them to the endpoint (a message driven bean).

You set the activation specification properties using the external service wizard and can change them using the IBM Integration Designer Assembly Editor, or after deployment through the administrative console.

The following table lists the activation specification properties. A complete 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 understanding property details.

Table 43. Activation specification properties

| Property name | | Description |
|--|-------------------------------|--|
| In the wizard | In the administrative console | |
| “Event groups to filter (AdapterInstanceEventFilter)” | adapterInstanceEventFilter | Controls whether this adapter instance processes specific event groups in the event store. |
| Ensure once-only event delivery (AssuredOnceDelivery) | assuredOnceDelivery | Specifies whether the adapter provides assured once delivery of events. |
| Delivery type | deliveryType | Determines the order in which events are delivered by the adapter to the export. |
| Event type filter | eventTypeFilter | A delimited list of event types that indicates to the adapter which events it should deliver. |
| Retry limit for failed events | failedEventRetryLimit | The number of times the adapter attempts to redeliver an event before marking the event as failed. |
| “Host name ” on page 140 | hostName | Specifies the name or address of the server that hosts Domino. |
| Interval between polling periods | PollPeriod | The length of time that the adapter waits between polling periods. |
| “Maximum events in polling period (PollQuantity)” on page 140 | PollQuantity | The number of events 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. |
| Minimum connections | minimumConnections | The minimum number of connections that the adapter can use for inbound event delivery. |
| Maximum number of retries in case of system connection failure | RetryLimit | The number of times the adapter tries to reestablish an inbound connection after an error. |
| “Password” on page 142 | password | Specifies the password of the user account of the adapter in the Lotus Domino server. |
| “DIIOP port number” on page 142 | portNumber | Specifies the port number that the adapter uses to access the Lotus Domino server. |
| Time between retries in case of system connection failure (milliseconds) | RetryInterval | The length of time that the adapter waits between attempts to reestablish connection after an error during inbound operations. |
| Retry EIS connection on startup | RetryConnectionOnStartup | Controls whether the adapter retries the connection to the Lotus Domino server if it cannot connect at startup. |
| Stop the adapter when an error is encountered while polling | StopPollingOnError | Specifies whether the adapter stops polling for events when it encounters an error during polling. |
| User name | userName | Specifies the name of the user account of the adapter in the Lotus Domino server. |

Event groups to filter (AdapterInstanceEventFilter)

This property controls whether this adapter instance processes specific event groups in the event store.

Table 44. AdapterInstanceEventFilter

| | |
|----------------|---|
| Required | No |
| Default | No default value |
| Property type | String |
| Usage | <p>You can use this property to filter the events during event processing by the adapter. Using this property, you can set the event filtering only to the event groups that you have subscribed. This property helps you to perform load balancing on high-volume event types by allowing multiple adapter instances to process events of the same type. When load balancing is not required, a single adapter instance processes all events of a given type.</p> <p>The AdapterInstanceEventFilter property corresponds to the ConnectorId column of the event table. Therefore, when you specify this property, the adapter filters the event groups based on your subscription.</p> <p>Note:</p> <ul style="list-style-type: none"> • All events are filtered when there is no value in the AdapterInstanceEventFilter property even when the value is set in the ConnectorId column of the event table. • If the EventTypeFilter and AdapterInstanceEventFilter properties are both set, the adapter processes only events that meet both criteria. That is, it processes only those events whose type is specified in the EventTypeFilter property and whose ConnectorId column matches the AdapterInstanceEventFilter property. |
| Example | Domi no001 |
| Globalized | No |
| Bidi supported | No |

Ensure once-only event delivery (AssuredOnceDelivery)

This property specifies whether to provide ensure once-only event delivery for inbound events.

Table 45. Ensure once-only event delivery details

| | |
|-----------------|--|
| Required | Yes |
| Possible values | True False |
| Default | True |
| Property type | Boolean |
| Usage | <p>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.</p> <p>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.</p> <p>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.</p> |
| Globalized | No |
| Bidi supported | No |

Delivery type (DeliveryType)

This property specifies the order in which events are delivered by the adapter to the export.

Table 46. Delivery type details

| | |
|-----------------|---|
| Required | No |
| Possible values | ORDERED UNORDERED ORDEREDBYKEY |
| Default | ORDERED |
| Property type | String |
| Usage | The following values are supported: <ul style="list-style-type: none">• ORDERED: The adapter delivers events to the export one at a time.• UNORDERED: The adapter delivers all events to the export at once.• ORDEREDBYKEY: The adapter delivers all events simultaneously except for the matching event keys. The events with matching keys are delivered to the export one at a time. |
| Globalized | No |
| Bidi supported | No |

Event types to process (EventTypeFilter)

This property contains a delimited list of event types that indicates to the adapter which events it must deliver.

Table 47. Event types to process details

| | |
|-----------------|---|
| Required | No |
| Possible values | A comma-delimited (,) list of business object types that can accept any one of the following values. <ul style="list-style-type: none">• EventTypeFilter=dbpath• EventTypeFilter=dbpath:form• EventTypeFilter=dbpath1:form1,dbpath2:form2• EventTypeFilter=dbpath1:form1:Operation1 Operation2,dbpath2:form2:Operation2 Operation3 |
| Default | All selected objects |
| Property type | String |

Table 47. Event types to process details (continued)

| | |
|----------------|---|
| Usage | <p>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 is applied and that all events are delivered to the endpoint.</p> <p>Note: The adapter delivers all the events automatically only when running the external service wizard. When modifying the artifacts through iterative development, ensure that you manually add the new events in the <i>Event type filter</i> field. The default value with no operations shown means that all supported operations are filtered by the adapter.</p> <p>The following parameters show the usage format for the <i>Event type filter</i> field.</p> <p>DBName:FormName:Operation1 Operation2, DBName:FormName:Operation1 Operation2 Operation3</p> <ul style="list-style-type: none"> • The comma (,) is used for separating the business objects. The adapter ignores objects that are excluded from this list. • The colon (:) is used for separating the database, business object, and the operation names. • The vertical bar () is used for separating the supported operations (such as Create, Delete, and Update). |
| Example | mail/report.nsf:newReport:Create Delete,mail/sample.nsf:newSample:Update |
| Globalized | No |
| Bidi supported | Yes |

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 48. Retry limit for failed events details

| | |
|-----------------|--|
| Required | No |
| Possible values | Integers |
| Default | 5 |
| Property type | Integer |
| Usage | <p>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:</p> <p>Default If this property is not set, the adapter tries five additional times before marking the event as failed.</p> <p>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.</p> <p>> 0 For integers greater than zero, the adapter retries the specified number of times before marking the event as failed.</p> <p>< 0 For negative integers, the adapter does not retry failed events.</p> |
| Globalized | No |
| Bidi supported | No |

Host name

This property specifies the name or address of the server that hosts Domino.

Table 49. Host name details

| | |
|----------------|---|
| Required | Yes |
| Default | No default value |
| Property type | String |
| Usage | Identifies the server, either by name or IP address, that hosts Domino. |
| Example | 9.26.159.264 |
| Globalized | No |
| Bidi supported | No |

Interval between polling periods (PollPeriod)

This property specifies the length of time that the adapter waits between polling periods.

Table 50. 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 |
| 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 |

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 51. 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 might improve slightly. |
| 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 52. Maximum 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. |
| 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 53. 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. |
| 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 54. 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 Lotus Domino 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 Lotus Domino server when it is first started, use the RetryConnectionOnStartup property. |
| Globalized | No |
| Bidi supported | No |

Password

This property specifies the password of the user account of the adapter in the Lotus Domino server.

Table 55. Password details

| | |
|----------------|---|
| Required | No |
| Default | No default value |
| Property type | String |
| Usage | The restrictions (case, length, and character) are determined based on the version of Domino. |
| Globalized | No |
| Bidi supported | Yes |

IIOP port number

This property specifies the port number the adapter uses to access the Lotus Domino server.

Table 56. Port number details

| | |
|----------------|--|
| Required | Yes |
| Default | 63148 |
| Property type | Integer |
| Usage | The port number is required when you use the external service wizard to discover objects and services. |
| Example | 9000 |
| 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 reestablish a connection.

Table 57. 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 | No |
| Bidi supported | No |

Retry EIS connection on startup (RetryConnectionOnStartup)

This property controls whether the adapter attempts to connect again to the Lotus Domino server if it cannot connect at startup.

Table 58. Retry EIS connection on startup details

| | |
|-----------------|--|
| Required | No |
| Possible values | True False |
| Default | False |
| Property type | Boolean |
| Usage | <p>This property indicates whether the adapter should retry the connection to the Lotus Domino server if the connection cannot be made when the adapter is started:</p> <ul style="list-style-type: none">• Set the property to False when you want immediate feedback about whether the adapter can establish a connection to the Lotus Domino 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 |

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 59. Stop the adapter when an error is encountered while polling details

| | |
|-----------------|---|
| Required | No |
| Possible values | True False |
| Default | False |
| Property type | Boolean |
| Usage | <p>If this property is set to True, the adapter stops polling when it encounters an error.</p> <p>If this property is set to False, the adapter logs an exception when it encounters an error during polling and continues polling.</p> |
| Globalized | No |
| Bidi supported | No |

User name

This property specifies the name of the user account, the adapter uses in the Lotus Domino server.

Table 60. User name details

| | |
|----------------|---|
| Required | No |
| Default | No default value |
| Property type | String |
| Usage | The restrictions (case, length, and character) are determined based on the Lotus Domino server version. |
| Example | DV1 |
| Globalized | No |
| Bidi supported | Yes |

Related concepts

“Inbound processing” on page 10

WebSphere Adapter for Lotus Domino supports inbound processing of events. The adapter polls the event database in the Lotus Domino server for events at specified intervals. Each time a form is created or updated, the adapter tracks the process as an event. When the adapter detects an event, it requests a copy of the form, converts the form data into a business object, and sends it to the consuming service.

“WebSphere Application Server environment variables” on page 22

WebSphere Application Server environment variables can be used in the external service wizard to specify directory values. You can now change any string property values in inbound and outbound configuration by only changing the environment variables.

Related tasks

“Setting deployment properties for a runtime environment” on page 63

The connection properties are initialized to the values that the wizard used to connect to the database. To configure the module to use other values, change values such as user name and password here.

“Setting location properties for artifacts” on page 69

After you select and configure deployment properties, use the external service wizard to configure location properties that the adapter uses to generate the service.

“Defining WebSphere Application Server environment variables” on page 41

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

Deployment and configuration of the WALD inbound extension manager

The adapter provides for deployment and configuration of the WALD inbound extension manager on the Lotus Domino server. When you deploy the extension manager, the adapter is able to monitor the create, update, and delete events in the Domino database.

Deployment of the inbound extension manager

You can deploy the inbound extension manager on Windows, AIX, Linux, and Solaris systems. After you deploy the required files, you can monitor the events in the specified databases in the Lotus Domino server.

The following steps describe how to deploy the required files for extension manager on the different operating systems.

1. Copy the required library file into the Lotus Domino server installation directory. See Table 1 to know the operating system-specific library file name.

Table 61. Operating system-specific library file

| Library file name | Supported operating systems |
|--------------------|-----------------------------|
| waldinbound.dll | Windows |
| libwaldinbound_r.a | AIX |
| libwaldinbound.so | Linux Solaris |

2. Edit the notes.ini file in your Domino directory and add the required line as given in Table 2.

Table 62. Add the extension manager line as per the operating system

| Extension manager lines | Supported operating systems |
|----------------------------------|-----------------------------|
| EXTMGR_ADDINS=waldinbound | Windows |
| EXTMGR_ADDINS=libwaldinbound_r.a | AIX |
| EXTMGR_ADDINS=libwaldinbound.so | Linux Solaris |

3. Copy the waldinbound.nsf file into your Domino data directory.
4. Restart the Lotus Domino server.

Note: You must restart the Lotus Domino server each time you update these files:

- waldinbound.dll
 - libwaldinbound_r.a
 - libwaldinbound.so
 - waldinbound.nsf
5. Click **Configuration for WALD Inbound** in the left pane to configure the databases that are to be monitored.
 6. Click **EventView** in the left pane to see the generated events.

Configuration of the inbound extension manager

You can configure the inbound extension manager by setting up different configuration parameters. The steps here describe the process of setting up the configuration parameters.

1. Open the waldinbound.nsf database in the Lotus Domino server through Lotus Notes®.
2. Click **EventTriggerConfigurationView**. You can create or edit an event trigger configuration view to configure a monitored Form.

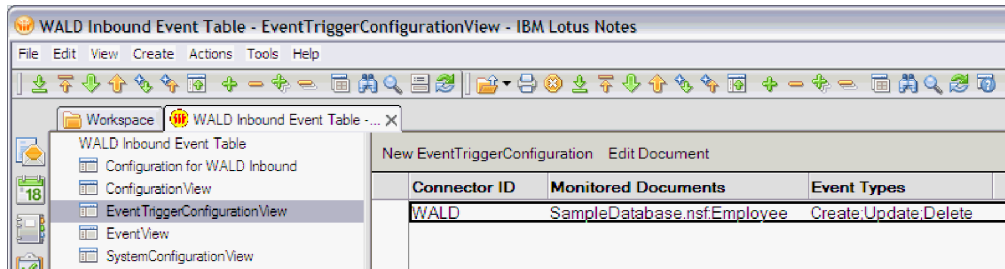


Figure 38. Editing the WALD inbound event trigger configuration

3. Click **Edit Document** to edit the existing event trigger configuration of the WALD inbound extension manager. The parameters available for configuration are described in Table 3.

Note: The event trigger configuration overwrites the **Connector ID**, **Database Paths**, and **Event Types** fields of configuration document.

Table 63. Event trigger configuration parameters for WALD inbound extension manager

| Configuration parameter | Description |
|-------------------------|---|
| Connector ID | The connector ID item that is used as the value of Connector ID field of the generated event. |
| Database Paths | The database paths and forms that are monitored by the WALD inbound extension manager. The semicolon ; is used as a separator between different configuration values. You can use either comma or semicolon as a separator to add another configuration value. The colon : is used as a separator between a database path and a form. For example, SampleDatabase.nsf:Employee;PlanDatabase.nsf implies that the form Employee in the database, SampleDatabase.nsf , and the whole database PlanDatabase.nsf are monitored. |
| Event Types | The event types that are monitored by the WALD inbound extension manager. The event types can be: <ul style="list-style-type: none"> • Create • Update • Delete You can configure the event monitoring by selecting the required events types. |

4. Click **SystemConfigurationView** to edit the existing logging configuration.

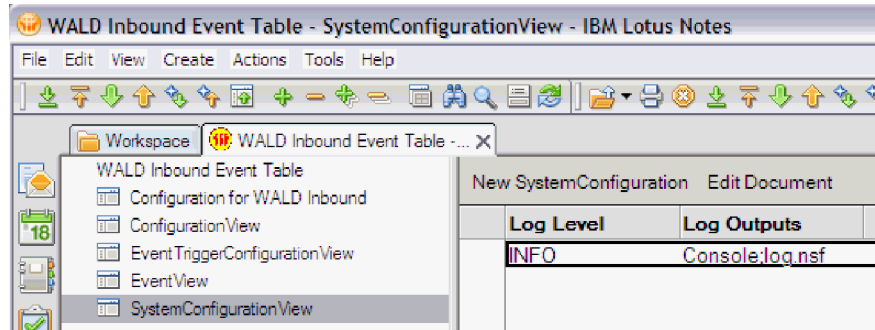


Figure 39. Editing the WALD inbound logging configuration

- Click **Edit Document** to edit the existing system configuration of the WALD inbound extension manager. The parameters available for configuration are described in Table 4.

Note: The values set in the system configuration overwrite the **Log Level** field of configuration document.

Table 64. Log configuration parameters for WALD inbound extension manager

| Configuration parameter | Description |
|-------------------------|---|
| Log level | The type of information that is logged during the WALD inbound processing. The log level can be: <ul style="list-style-type: none"> • NONE • ERROR • INFO • DEBUG |
| Log outputs | The targets where you can log the information. The log can be logged to the following targets: <ul style="list-style-type: none"> • Console • log.nsf |

- Click **ConfigurationView**.

Note: When both event trigger configuration documents and configuration document are available, the adapter uses the event trigger configuration document to trigger the events.

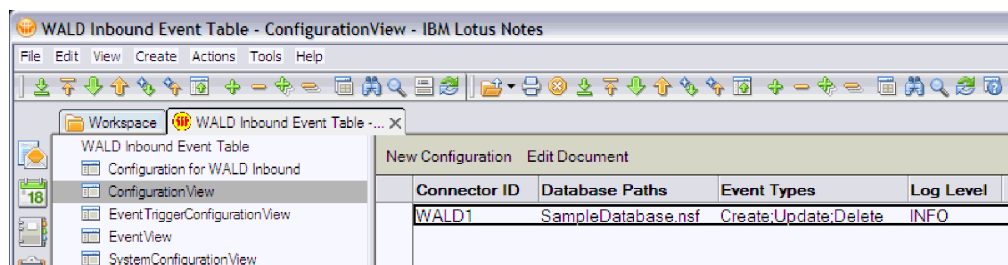


Figure 40. Editing the WALD inbound event configuration

7. Click **Edit Document** to edit the existing configuration of the WALD inbound extension manager. The parameters available for configuration are described in Table 5.

Note: The configuration done using the **ConfigurationView** is a configuration document used only for WALD version 7.0. When either **EventTriggerConfigurationView** or **SystemConfigurationView** is configured, the existing configuration values of configuration document are overwritten.

Table 65. Configuration parameters for WALD inbound extension manager

| Configuration parameter | Description |
|-------------------------|---|
| Connector ID | The connector ID item that is used as the value of Connector ID field of the generated event. |
| Database Paths | The database paths and forms that are monitored by the WALD inbound extension manager. The semicolon ; is used as a separator between different configuration values. You can use either comma or semicolon as a separator to add another configuration value. The colon : is used as a separator between a database path and a form. For example, SampleDatabase.nsf:Employee;PlanDatabase.nsf implies that the form Employee in the database, SampleDatabase.nsf and the whole database PlanDatabase.nsf are monitored. |
| Event Types | The event types that are monitored by the WALD inbound extension manager. The event types can be: <ul style="list-style-type: none"> • Create • Update • Delete You can configure the event monitoring by selecting the required events types. |
| Log level | The type of information that is logged during the WALD inbound processing. The log level can be: <ul style="list-style-type: none"> • NONE • ERROR • INFO • DEBUG |

8. Click **Configuration for WALD Inbound** to view all the configuration documents.

Note: The event trigger configuration and system configuration overwrite the configuration document of WALD version 7.0.

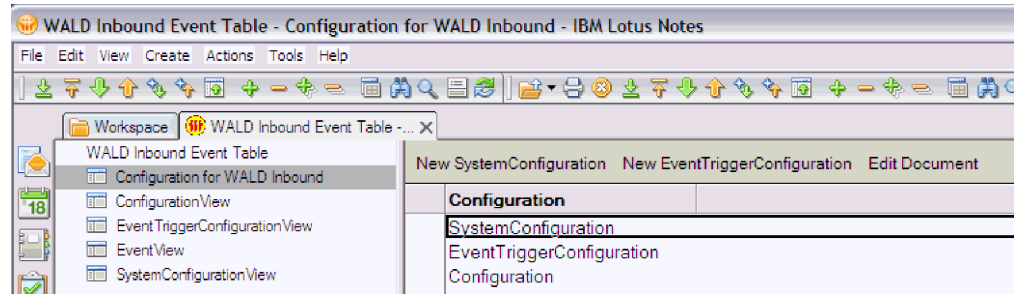


Figure 41. Viewing the WALD inbound configuration tables

9. Save the document. The configuration settings take effect immediately after you save the configuration document.

Related concepts

“Inbound extension manager” on page 14

During inbound processing, WebSphere Adapter for Lotus Domino (WALD) inbound extension manager monitors the operations on the Lotus Domino server. Whenever you create, update, or delete a document from the Domino database, the inbound extension manager generates corresponding events containing the operation information.

View and verify the deployment of the WALD inbound extension manager

After you have deployed the inbound extension manager, you can view and verify its deployment to check if the extension manager is correctly configured to monitor the events in the Lotus Domino server.

View the events

The following steps describe how to view the generated events of the waldinbound.nsf database.

1. Open the application (database) waldinbound.nsf in the Lotus Domino server through Lotus Notes.
2. Click **EventView** in the left pane to see the list of all the generated events before the events are fetched by a run time inbound application using the WebSphere Adapter for Lotus Domino.

Verify the deployment

You can verify the deployment of the extension manager after deploying the dependency files (waldinbound.dll, libwaldinbound_r.a, or libwaldinbound.so) and waldinbound.nsf files on the Lotus Domino server. To verify the deployment, follow these steps:

1. Open the application (database) waldinbound.nsf in the Lotus Domino server through Lotus Notes.
2. Click **SystemConfigurationView** in the left pane.
3. Click **Edit Document** to edit the system configuration document.

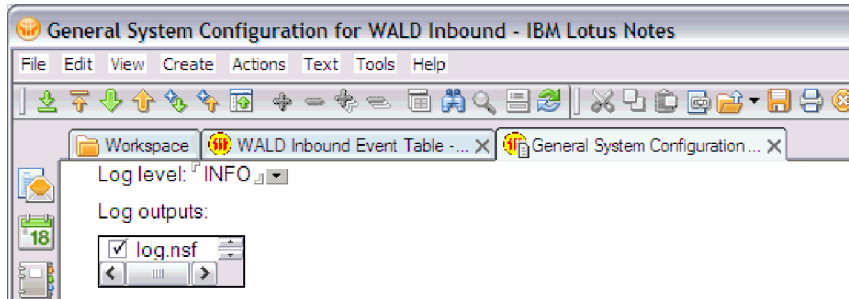


Figure 42. Editing the WALD inbound event table logging configuration

4. Change the **Log Level** to INFO or DEBUG level.
5. Check the log messages by reading the **Configuration Profile** printed on the Lotus Domino server console.

Related concepts

“Inbound extension manager” on page 14

During inbound processing, WebSphere Adapter for Lotus Domino (WALD) inbound extension manager monitors the operations on the Lotus Domino server. Whenever you create, update, or delete a document from the Domino database, the inbound extension manager generates corresponding events containing the operation information.

Globalization

WebSphere Adapter for Lotus Domino 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.

Related tasks

“Setting deployment properties for a runtime environment” on page 52

The connection properties are initialized to the values that the wizard used to connect to the database. To configure the module to use other values, change values such as user name and password here.

“Setting location properties for artifacts” on page 56

After you select and configure deployment properties, use the external service wizard to configure location properties that the adapter uses to generate the service.

Support for globalization

The adapter is globalized to support single and multibyte character sets and deliver message text in the specified language.

Introduction

Globalized software applications are designed and developed for use within multiple linguistic and cultural environments rather than a single environment. WebSphere Adapters, IBM Integration Designer, IBM Business Process Manager, 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 encoding for characters in most known

character code sets (both single and multibyte). Therefore, when data is transferred between these integration system components, there is no need for character conversion.

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 globalization for business objects and adapter foundation classes (AFC) related properties.

Globalization support for outbound and inbound properties

The WebSphere Adapter for Lotus Domino supports globalized character for the following outbound and inbound properties.

- Database name
- Database path
- Form name
- Form fields
- Form attributes
- Business data (values for the form attributes)

Related reference

“Activation specification properties” on page 135

Activation specification properties are properties that hold the inbound event processing configuration information for a message endpoint.

“Managed (J2C) connection factory properties” on page 121

Managed connection factory properties are used by the adapter at run time to create an outbound connection instance with the Lotus Domino server.

Adapter messages

View the messages issued by WebSphere Adapter for Lotus Domino at the following location.

Link to messages: http://publib.boulder.ibm.com/infocenter/dmndhelp/v7r5mx/topic/com.ibm.wbpm.ref.doc/topics/welc_ref_msg_wbpm.html

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 CWYDM are issued by WebSphere Adapter for Lotus Domino
- 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 WebSphere Adapter for Lotus Domino.

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

- IBM Business Process Manager, version 7.5, information center, which includes IBM Business Process Manager, IBM WebSphere Enterprise Service Bus, and IBM Integration Designer information: <http://publib.boulder.ibm.com/infocenter/dmndhelp/v7r5mx/index.jsp>.
- IBM Business Process Manager, version 7.0, information center, which includes IBM Business Process Manager, IBM WebSphere Enterprise Service Bus, and IBM Integration Designer information: <http://publib.boulder.ibm.com/infocenter/dmndhelp/v7r0mx/index.jsp>.
- WebSphere Adapters, version 6.2.x, information center: <http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r2mx/index.jsp>.
- IBM WebSphere Adapters, version 7.5 installation on WebSphere Application Server, version 8.0 information: <http://www-01.ibm.com/support/docview.wss?rs=695&uid=swg27011040>.

developerWorks® resources

- WebSphere Adapter Toolkit
- WebSphere business integration zone

Video samples

To help you use WebSphere Adapters, sample videos are available to detail some of the scenarios.

You can find several Adapter video demos at this FTP location <ftp://ftp.software.ibm.com/software/websphere/integration/wsa/library/videos/>.

To view these video samples:

1. Download the zip file of the required sample from the FTP location.
2. Extract the downloaded zip file to your system.
3. From the extracted folder, double-click the HTML file.
4. The video sample will open in your default Web browser.

Note: The video has a built in audio narration. If you do not want to listen to the audio, you can lower its tab to the minimum, and slide the tab to an audio level, whenever you want the audio back.

The following demos are currently available at this location: <ftp://ftp.software.ibm.com/software/websphere/integration/wsa/library/videos/>

- AFC: Scheduling Calendar based polling
- AFC: Overcoming an AFC version conflict situation
- Lotus Domino: Configuring notification rules for forms in database paths, and generating their events
- IBM i: Supporting CCSID in the RPG, COBOL, and service programs
- IBM i: Discovering the PCMLS in a local file system

- JDE: Retrieving data using multiple business functions in outbound processing
- Oracle: Integrating support for an Oracle AQ activity
- Oracle: Integrating an XML gateway through JMS in Oracle E-Business Suite
- PeopleSoft: Handling a RecordNotFound fault behavior
- PeopleSoft: Handling an event filtering
- SAP: Monitoring Tivoli ITCAM for a BPM Portfolio requirement
- SAP: Adding random numbers to the end of duplicate field names
- Siebel: Adding codepage properties to both EMD and runtime for an inbound Business Object scenario
- Siebel: Supporting more than one output argument for a Siebel Business Services Interface in an outbound scenario
- WAT: Creating a custom adapter using WebSphere Adapter Toolkit
- WAT: Creating a custom adapter outbound interface in WebSphere Adapter Toolkit
- WAT: Testing a module in WebSphere Adapter Toolkit

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](http://www.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
 - setting in administrative console 91, 96
- Activation specification properties 135
 - event delivery 135
 - events
 - filtering 135
 - list of 135
 - maximum connection retry 135
 - minimum connection retry 135
 - poll period 135
 - port number
 - DIOP 135
 - retry connection 135
 - user account
 - password 135
 - user name 135
- adapter 44
 - information resources 151
 - messages 151
 - package files 103
 - performance 98
 - project, create 41
 - related information 151
 - samples and tutorials 151
 - support and assistance 151
 - technotes 151
- adapter application
 - starting 97
 - stopping 97
- Adapter for Lotus Domino module
 - exporting as EAR file 83
 - installing EAR file on server 84
 - starting 97
 - stopping 97
- adapter implementation
 - planning 25
 - security 25
- alias, authentication 38
 - password
 - clear text 38
- application
 - communicating
 - two-way 1
- artifacts
 - import, XSD, WSDL
 - storing 56
 - location properties
 - Setting 56, 69
- attachments
 - FilePath 4
- attribute properties 111, 112, 113
- authentication
 - description 26
 - in the wizard 26
 - runtime environment 26
- authentication alias 38
 - J2C 26

B

- browse 107, 108
- business faults 21, 114, 145
- business faults behavior
 - run time
 - setting 121
- business graph 12
- Business object attribute type 111
- Business object attributes
 - mapping 111
- business object structure 17
 - form-based business object structure
 - Universal ID 17
- business object, predefine 40
- Business objects
 - attributes 111, 112, 113
 - how to view 48
 - naming conventions 114
 - supported operations 4
 - viewing 60

C

- CEI (Common Event Infrastructure) 101
- Common Event Infrastructure (CEI) 101
- compatibility matrix 2
- complex item 111
 - data type
 - field type 111
- Composite properties
 - inbound
 - specifying 62
 - outbound
 - specifying 50
- confidential data, disguising 25
- confidential tracing 25
- configuration overview 37
- configuration properties
 - inbound 131
- ConfigurationView 14
- configuring
 - logging properties 103
 - Performance Monitoring Infrastructure (PMI) 98
- connection factory
 - managed (J2C) 121
- connection failure
 - retrying
 - maximum, minimum 121
- connection properties
 - specifying 52
- Connection properties 52
- Connector ID 14
- connector project 41
- Create operation
 - document
 - Universal ID 4
- custom properties
 - activation specification 91, 96
 - managed connection factory 89, 94

- custom properties (*continued*)
 - resource adapter 87, 93

D

- database 107, 108
 - format 2
- debugging
 - self-help resources 109
- default namespace 56
- Delete operation
 - document
 - Universal ID 9
- DeleteResponse business object
 - structure 17
- deployment 84
 - production environment 82
 - test environment 79
- deployment environment 79
- deployment options 26
- Deployment properties
 - runtime environment
 - setting 64
 - setting 52, 63
- Deployment Properties
 - runtime environment
 - setting 52
- developerWorks 152
- disguising confidential data 25
- Document 2
 - updating 5
- Document business data 6
- Domino
 - client
 - Administrator, Designer 2
- Domino Internet Inter-ORB Protocol (DIOP) port number 3
 - specifying 121

E

- EAR file
 - exporting 83
 - installing on server 84
- earlier versions
 - compatibility 30
- edit binding
 - import component 73, 74
- embedded adapter
 - activation specification properties,
 - setting 91
 - considerations for using 28
 - managed connection factory
 - properties, setting 89
 - resource adapter properties,
 - setting 87
 - usage considerations 26
- embedded adapters
 - changing configuration properties 87

- embedded adapters (*continued*)
 - setting activation specification properties 91
 - setting managed (J2C) connection factory properties 89
 - setting resource adapter properties 87
- environment variable
 - scope
 - server, node, or cell 22
 - specifying 22
- event based filtering 10
- event business components 12
- event delivery 138
- event recovery mechanism 14
- event store
 - event business components 12
 - event record 12
 - event triggers 12
 - polling 12
- events
 - description 149
- EventTriggerConfigurationView 14
- exception
 - server access
 - denying 108
- Exception
 - org.omg.CORBA.NO_PERMISSION 107
- exists 107
- Exists operation
 - document
 - Universal ID 6
 - WBIFault 21
- ExistsResponse business object
 - structure 17
- exporting module as EAR file 83
- extension manager 10
- external service connection
 - properties 117, 131
- external service wizard 20
 - connection properties 45
 - setting 45
 - start up 45

F

- faults
 - description 21, 114, 145
- files
 - SystemOut.log log file 105
 - trace.log trace file 105
- foreign key 111
- Form 2
- Form-based business object structure 17

G

- Globalization 150
 - outbound, inbound properties supporting 150

H

- hardware and software requirements 2
- hardware requirements 2

- High Availability (HA)
 - clustered environments 151
- Host name 45
 - Internet, system 107
 - server 121
- HTTP 107, 108

I

- IBM Business Process Manager
 - information 152
- IBM Business Process Manager or WebSphere Enterprise Service Bus
 - deploying to 82
- IBM Business Process Manager, version 7.0, information 152
- IBM i adapter 44
- IBM Integration Designer
 - information 152
 - starting 40
 - test environment 79
- IBM WebSphere Adapter for Lotus Domino
 - administering 87
- IBM WebSphere Adapter Toolkit 152
- developerWorks resources 151
- IBM WebSphere Enterprise Service Bus
 - information 152
- idle session 107
- illegal XML characters
 - processing 52, 64
- implementation, Java 80
- import component 73, 74
- inbound
 - configuration properties 131
- Inbound event processing 10
- inbound extension manager 14
 - deploying, configuring 145
- inbound processing 44
- Inbound processing 10
 - discovering
 - form fields 58
 - querying 58
- installing EAR file 84
- interaction specification properties
 - changing 77
- interaction specification property 127
- Item 2
- iterative development
 - connection-based editing 73, 74
 - Edit Binding 74
 - import component 73, 74

J

- J2C authentication data entry 52
- Java 2 security 25, 26
- Java implementation 80

L

- Library file
 - Supported operating systems 145
- Library file name 14
- Log Analyzer 103

- log and trace
 - configure 103
- Log and Trace Analyzer, support for 20
- log and trace files 20
- log files
 - changing file name 105
 - disabling 103
 - enabling 103
 - level of detail 103
 - location 105
 - SystemOut.log 105
- logging
 - configuring properties with
 - administrative console 103
- Logging and tracing
 - configuring 64
- logging level 103
- Logging level 45
- logging properties
 - change 45
- Lotus Domino Administrator client 2
- Lotus Domino Designer client 2
- Lotus Domino driver files 20, 44
- Lotus Domino server 2

M

- managed (J2C) connection factory
 - properties
 - setting in administrative console 89, 94
- matrix, compatibility 2
- message text
 - appropriate language
 - delivering 150
- messages, adapter 151
- migration
 - performing migration 31
 - upgrading a project 33
- Migration considerations 30
- migration wizard
 - running 30
- module
 - adding to the server 80
 - configuring for deployment
 - overview 37
 - configuring inbound processing 58
 - configuring outbound processing 46
 - deploy for testing 79
 - deploying
 - applications 52
 - module, creating 40
 - monitoring performance 98
 - multiple adapters
 - installing 108
 - multiple connection 138

N

- naming conventions for business
 - objects 114
- No data set 6
- Notes exception 107, 108

O

- object 107
- operations
 - create 4
 - delete 9
 - exists 6
 - retrieve 9
 - retrieveAll 6
 - update 5
- outbound configuration properties 117
- outbound fault handling 114
- Outbound operations 4
- outbound processing 44
 - testing the module 81
- Outbound processing 3

P

- package files for adapters 104
- ParentDocumentUNID 6
- Password
 - specifying 121
- performance monitoring
 - infrastructure 98
 - configuring 98
 - performance statistics 100
- Performance Monitoring Infrastructure (PMI)
 - configuring 98
 - description 98
 - viewing performance statistics 100
- performance statistics 100
- PMI (Performance Monitoring Infrastructure)
 - configuring 98
 - viewing performance statistics 100
- PMI events
 - identifying 121
- polling 12
- problem determination 108
 - self-help resources 109
- process 44
- Project
 - multiple adapters
 - Migrating 30
- project interchange (PI) file
 - project interchange files 33
 - projects 33
 - updating without migrating 33
- properties
 - activation specification 91, 96
 - list of 135
 - configuration properties
 - inbound 131
 - outbound 117
 - external service connection 117, 131
 - inbound configuration 131
 - managed (J2C) connection factory 89, 94
 - outbound configuration 117
 - resource adapter 87, 93
- properties information
 - guide 116

R

- RAR (resource adapter archive) file
 - description 82
 - installing on server 82
- recommended fixes 109
- record, event 12
- Redbooks, IBM WebSphere Adapters 151
- related products, information 151
- request 44
- requirements
 - hardware 2
 - software 2
- resource adapter archive (RAR) file
 - description 82
 - installing on server 82
- resource adapter properties
 - details 119, 133
 - setting in administrative console 87, 93
- response 44
- Retrieve operation 9
- RetrieveAll operation 6
 - maximum records 50
- Retry limit property 141
- RichText
 - managing attachments 17
- RichText business object structure 17
- RichTextComplexItem business object structure 17
- road map for configuring the module 37
- runtime environment 44
 - deploying EAR file 82

S

- samples 35, 152
- Search criteria 6
 - multiple fields
 - AND operator 6
 - multiple values 6
 - single values
 - numbers, date 6
 - text search for rich text contents 6
- security
 - disguising sensitive data 25
 - user authentication 26
- security features
 - adapter 25
 - Java 2 security 25
- security, Java 2 26
- selected objects
 - configuring 50
 - inbound
 - configuring 62
- self-help resources 109
- sensitive data, disguising 25
- service 44
- Service Component Architecture (SCA) 2
- service export 44, 45
- service import 44, 45
- software dependencies 20, 44
- software requirements 2
- stale connection
 - avoiding 106

- stand-alone adapter 96
 - considerations for using 28
 - managed connection factory
 - properties, setting 94
 - resource adapter properties,
 - setting 93
 - usage considerations 26
- stand-alone adapters
 - changing configuration properties 93
 - setting activation specification
 - properties 96
 - setting managed (J2C) connection factory properties 94
 - setting resource adapter properties 93
- starting adapter applications 97
- stopping adapter applications 97
- stores
 - common properties
 - metadata 17
- support
 - overview 103
 - plug-in for IBM support
 - assistant 109
 - self-help resources 109
 - web site 109
- Support for business faults during runtime 52
- Supported operations 4
- SystemConfigurationView 14
- SystemOut.log file 105

T

- target component 79
- technotes 2, 109
- test environment 79
 - adding module to 80
 - deploying to 80
 - testing modules 81
- timeout 107
- trace files
 - changing file name 105
 - disabling 103
 - enabling 103
 - level of detail 103
 - location 105
 - trace.log 105
- tracing
 - configuring properties with administrative console 103
- transaction ID (XID) 12
- triggers, event 12
- troubleshooting
 - overview 103
 - self-help resources 109
- tutorials 35, 152

U

- UNORDERED 138
- Update operation 5
 - document
 - Universal ID, Note ID 5
- upgrade
 - Deciding 30

- user data
 - disguising 121
- user name
 - run time 121

V

- version conflict 108

W

- WALD 14
- WebSphere Adapters, version 6.0,
 - information 152
- WebSphere Adapters, version 6.2.x,
 - information 152
- WebSphere Application Server
 - environment variables 22
- WebSphere Application Server
 - environment variables, defining 41
- WebSphere Application Server
 - information 152
- WebSphere Business Integration Adapters
 - information 152
- wiring components 79



Printed in USA