Artix Connect for WCF

BizTalk Integration Guide

Version 1.0
June 2008

Making Software Work Together™

BizTalk Integration Guide

IONA Technologies

Version 1.0

Published 04 Sep 2008

Trademark and Disclaimer Notice

IONA Technologies PLC and/or its subsidiaries may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this publication. Except as expressly provided in any written license agreement from IONA Technologies PLC, the furnishing of this publication does not give you any license to these patents, trademarks, copyrights, or other intellectual property. Any rights not expressly granted herein are reserved.

IONA, IONA Technologies, the IONA logo, Orbix, High Performance Integration, Artix, FUSE, and Making Software Work Together are trademarks or registered trademarks of IONA Technologies PLC and/or its subsidiaries.

Java and J2EE are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. CORBA is a trademark or registered trademark of the Object Management Group, Inc. in the United States and other countries. All other trademarks that appear herein are the property of their respective owners.

While the information in this publication is believed to be accurate, IONA Technologies PLC makes no warranty of any kind to this material including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. IONA shall not be liable for errors contained herein, or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.

Copyright Notice

Table of Contents

List of Figures

List of Tables

List of Examples

Preface

The Artix Connect for WCF Library

The Artix Connect for WCF documentation library consists of the following books:

- Installation Guide [http://www.iona.com/support/docs/artix/connectwcf/1.0/install_guide/index.html]

  Read the Installation Guide if you are about to install Artix Connect for WCF.

- Release Notes [http://www.iona.com/support/docs/artix/connectwcf/1.0/release_notes/index.html]

  Read the Release Notes for a list of features, known issues, and release-specific information.

- Getting Started Guide [http://www.iona.com/support/docs/artix/connectwcf/1.0/tutorial/index.html]

  Read this Getting Started Guide if you are new to Artix Connect for WCF and want to walk through a step-by-step tutorial that shows you how to use Artix Connect for WCF to integrate a .NET application with a CORBA and JMS back-end.

- User's Guide [http://www.iona.com/support/docs/artix/connectwcf/1.0/users_guide/index.html]

  Read the User's Guide if you want to use Artix Connect for WCF to integrate a .NET application with a back-end CORBA system or a back-end JMS queue or topic.

- BizTalk Integration Guide [http://www.iona.com/support/docs/artix/connectwcf/1.0/biztalk_guide/index.html]

  Read the BizTalk Integration Guide if you want to walk through a steps-by-step tutorial that shows you how to use Artix Connect for WCF to integrate BizTalk Server 2006 or BizTalk Server 2006 R2 with a JMS back-end system and a CORBA back-end system.

Document Conventions

Typographical conventions

This book uses the following typographical conventions:

`fixed width`	Fixed width (Courier font) in normal text represents portions of code and literal names of items such as classes, functions, variables, and data structures. For example, text might refer to the `javax.xml.ws.Endpoint` class. Constant width paragraphs represent code examples or information a system displays on the screen. For example: `import java.util.logging.Logger;`

`Fixed width`
`italic`
	Fixed width italic words or characters in code and commands represent variable values you must supply, such as arguments to commands or path names for your particular system. For example:
	`% cd /users/`*`YourUserName`*

Italic	Italic words in normal text represent *emphasis* and introduce *new terms*.
Bold	Bold words in normal text represent graphical user interface components such as menu commands and dialog boxes. For example, the **User Preferences** dialog.

Keying conventions

This book uses the following keying conventions:

No prompt	When a command's format is the same for multiple platforms, the command prompt is not shown.
>	The notation > represents the MS-DOS or Windows command prompt.
`...`	Horizontal or vertical ellipses in format and syntax descriptions indicate that material has been eliminated to simplify a discussion.
`[ ]`	Brackets enclose optional items in format and syntax descriptions.
`{ }`	Braces enclose a list from which you must choose an item in format and syntax descriptions.
`|`	In format and syntax descriptions, a vertical bar separates items in a list of choices enclosed in `{}` (braces).

Admonition conventions

This book uses the following conventions for admonitions:

🗎	Notes display information that might be useful, but not critical.
🔔	Tips provide hints about completing a task or using a tool. They may also provide information about workarounds to possible problems.
!	Important notes display information that is crucial to the task at hand.
▼	Cautions display information about likely errors that can be encountered. These errors are unlikely to cause damage to your data or your systems.
✖	Warnings display information about errors that might cause damage to your systems. Possible damage from these errors include system failures and loss of data.

Introduction

This chapter describes, at a high-level, how Artix Connect for WCF connects BizTalk Server 2006 or BizTalk Server 2006 R2 to CORBA and JMS external services.

What is Artix Connect for WCF?

Introduction

Artix Connect for WCF is a BizTalk Server 2006 WCF custom binding adapter that makes integrating BizTalk applications with external services, such as CORBA and JMS, simple and easy. It is implemented using the BizTalk Server Adapter Framework.

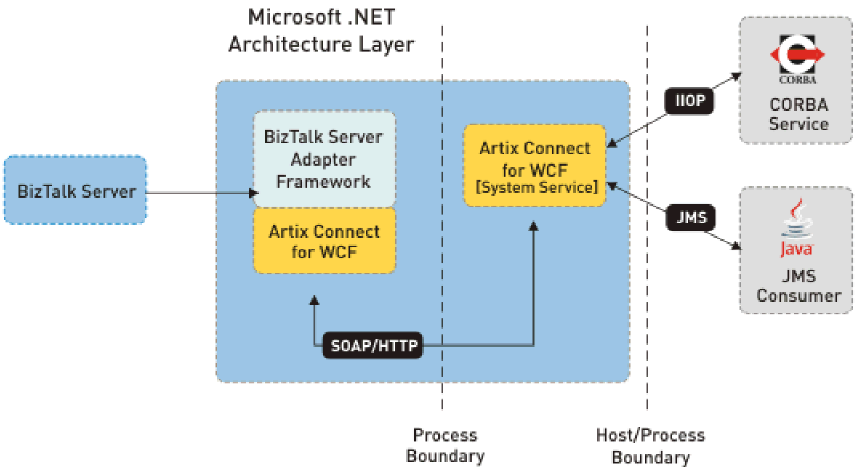
Artix Connect for WCF includes a wizard that, in a few steps, takes your target external service contact information and generates the XML schema definitions and binding metadata needed for BizTalk to contact that service. Using Artix Connect for WCF means that you do not have to manually create the binding information and configuration is made easy.

Artix Connect for WCF supports both one-way and two-way message exchange patterns.

Graphical representation

Figure 1 on page 16 shows Artix Connect for WCF's integration with BizTalk from an architecture point of view.

Figure 1. Artix Connect for WCF BizTalk Integration



How it works

Once you supply the Artix Connect for WCF wizard with the contact information for your external target service, it generates the XML schema definitions and binding metadata for that service. You simply use the schema and metadata to bind the target service endpoint to the BizTalk Send and Receive Ports when you are designing business process orchestrations. Using Artix Connect for WCF, you can:

- Configure Send Ports to consume CORBA and JMS services, thus enabling BizTalk to transmit messages to these external services.
- Expose endpoint metadata and configure Receive Locations to publish SOAP messages to the BizTalk MessageBox database.

Assumptions

This document assumes that you are familiar with BizTalk Server 2006 or BizTalk Server 2006 R2. It does not assume that you are familiar with Artix Connect for WCF, JMS or CORBA, although some knowledge would be helpful.

Running the Tutorial

This chapter walks you through a tutorial that shows you how to use Artix Connect for WCF to enable BizTalk Server to communicate with CORBA and JMS. It assumes that you are running either the CORBA or the JMS sample application at any one time. The steps are the same, but the details are specific to each sample application. The screenshots are taken from the CORBA sample application.

Introduction to CORBA Sample Application

CORBA sample application

The BizTalk CORBA sample application is located in the following directory of your Artix Connect for WCF installation:

```
InstallDir\Visual Studio Adapter\samples\biztalk\corba
```

It contains the following subfolders:

- `bin`—contains prebuilt executable for the CORBA service.
- `corba`—contains the source code for the CORBA system.
- `biztalk`—contains the Visual Studio 2005 solution for the BizTalk application.
- `etc`—contains the IDL file for the CORBA system and a configuration file.

Before you begin

Before you use Artix Connect for WCF to connect to a CORBA service, you must have

1. Access to the CORBA service IDL file.
2. Access to the CORBA object reference. Artix Connect for WCF supports the following object reference formats:
   i. `IOR` in the form of a string or a file
   ii. `corbaloc URL`
   iii. `corbaname URL`

These details are requested by the Artix Connect for WCF wizard and are provided for you for the sample application.

Introduction to CORBA

For an high-level introduction to CORBA, see the Introduction to CORBA in the *User's Guide*.

Introduction to JMS Sample Application

Overview

The BizTalk JMS sample application is located in the following directory of your Artix Connect for WCF installation:

```
InstallDir\Visual Studio Adapter\samples\biztalk\jms
```

It contains the following subfolders:

- `bin`—contains prebuilt executable for the JMS service.
- `biztalk`—contains the Visual Studio 2005 solution for the BizTalk application.
- `server`—contains the source code and Java class files for the JMS system.

Before you begin

Before you use Artix Connect for WCF to connect to a JMS queue or topic, you need to know:

- The JMS implementation of the system to which you are trying to connect.
- Connection details for the JMS broker you are using.
- Details of the queue or topic that you are using.

These details are requested by the Artix Connect for WCF wizard and are provided for you for the sample application.

Introduction to JMS

For an high-level introduction to JMS, see the Introduction to JMS in the *User's Guide*.

Step 1: Setting Configure for BizTalk Option.

Introduction

Important

Before using Artix Connect for WCF with BizTalk, you must ensure that the **Configure for BizTalk** option in the Artix Administration tool is selected.

Steps

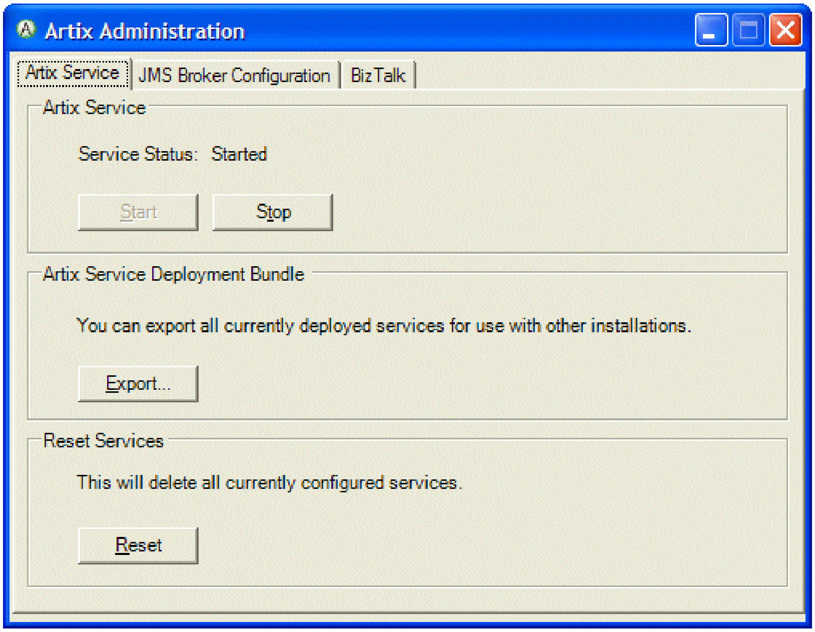
To set the Configure for BizTalk option, complete the following steps:

1. Launch the Artix Administration tool from the Windows start menu:

   (All) Programs | IONA | Artix Connect For WCF | Artix Administration

   It appears as shown in Figure 2 on page 22.

   Figure 2. Artix Administration Tool



Service Status: Started

You can export all currently deployed services for use with other installations.

This will delete all currently configured services.

2. Select the BizTalk tab.

3. Tick the **Configure for BizTalk** check box.

4. Close the Artix Administration tool.

Step 2: Running the Back-end Services

Overview

The sample applications include back-end services. To run the:

CORBA sample: Run the CORBA service.

JMS sample: Run the JMS broker and the Java server.

Running the CORBA service

To run the CORBA service:

1. Open a Windows Explorer window and navigate to the following directory of your Artix Connect for WCF installation:

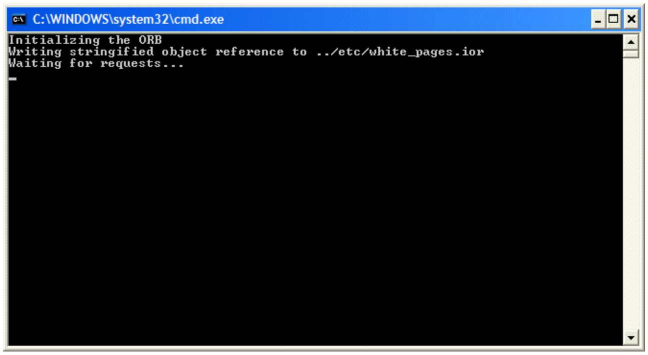
   ```
   InstallDir\Visual Studio Adapter\samples\biztalk\corba\bin
   ```

2. Double-click on the `start_corba_server.bat` file to start the CORBA server.

3. If the Windows Firewall asks if you want to unblock the application, select **Unblock**.

The CORBA server takes a few seconds to start. When it is ready and listening for incoming requests, it should appear as shown in Figure 3 on page 25.

Figure 3. CORBA Server Ready and Waiting for Requests



```
Initializing the ORB
Writing stringified object reference to ../etc/white_pages.ior
Waiting for requests...
_
```

Running the JMS broker

The sample application runs by default against either Apache ActiveMQ or FUSE Message Broker (which is an open source JMS broker based on Apache ActiveMQ).

If you do not already have a JMS broker, download and install FUSE Message Broker from the following website:

http://open.iona.com/downloads

Install using all of the default settings.

To run FUSE Message Broker:

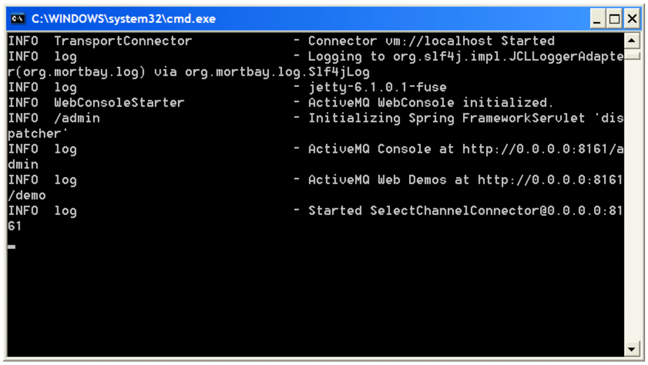
1. Open a Windows Explorer window and navigate to the following directory of your FUSE Message Broker or Apache ActiveMQ installation:

   ```
   InstallDir\bin
   ```

2. Double-click on the `activemq.bat` file to start the JMS broker.

3. Wait for the broker to fully initialize; that is, until you see the message `"Started SelectChannelConnector@0.0.0.0:8161"` (see Figure 4 on page 26).

Figure 4. Fully Initialized FUSE Message Broker



```
INFO  TransportConnector                 - Connector vm://localhost Started
INFO  log                                - Logging to org.slf4j.impl.JCLLoggerAdapte
r(org.mortbay.log) via org.mortbay.log.Slf4jLog
INFO  log                                - jetty-6.1.0.1-fuse
INFO  WebConsoleStarter                  - ActiveMQ WebConsole initialized.
INFO  /admin                             - Initializing Spring FrameworkServlet 'dis
patcher'
INFO  log                                - ActiveMQ Console at http://0.0.0.0:8161/a
dmin
INFO  log                                - ActiveMQ Web Demos at http://0.0.0.0:8161
/demo
INFO  log                                - Started SelectChannelConnector@0.0.0.0:81
61
```

Running the Java server

Important

To run the sample Java server add the JMS implementation JAR to your `CLASSPATH` and run the `start_java_server.bat` file as described below:

1. Open a Windows command prompt and run the following command:

Example 1. Starting Java Server

```
set CLASSPATH=MyBroker.jar;%CLASSPATH%
InstallDir\Visual Studio Adapter\samples\biztalk\jms\bin\start_java_server.bat
```

For FUSE Message Broker 5.0.0.9, for example, run:

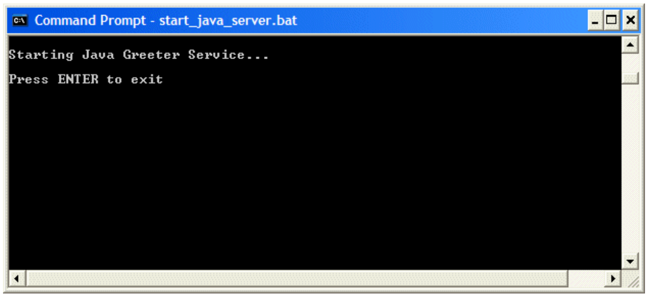
Example 2. FUSE Message Broker: Starting Java Server

```
set CLASSPATH=FUSEInstallDir\activemq-all-5.0.0.9-fuse.jar;%CLASSPATH%
InstallDir\Visual Studio Adapter\samples\biztalk\jms\bin\start_java_server.bat
```

For a list of JMS implementation JARs for all supported JMS brokers, see JMS Broker Implementation JARs in the *Installation Guide*:

2. Wait for the Java service to fully initialize and connect to the JMS broker; that is, until it displays the message `"Press ENTER to exit"` (see Figure 5 on page 27).

Figure 5. Fully Initialized Java Server



```
Starting Java Greeter Service...
Press ENTER to exit
```

Step 3: Opening the BizTalk Solution

Steps

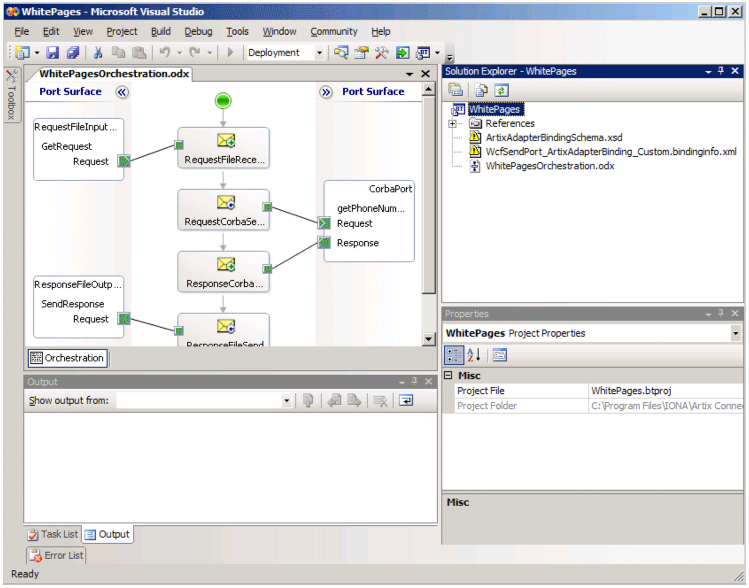
To open the BizTalk solution:

1. Open a Windows Explorer window and navigate to the following directory of your Artix Connect for WCF installation:
   - **CORBA sample:** `*InstallDir*\Visual Studio Adapter\samples\biztalk\corba\biztalk`.
   - **JMS sample:** `*InstallDir*\Visual Studio Adapter\samples\biztalk\jms\biztalk`.

2. Double-click on the `.sln` solution file.
   - **CORBA sample:** `WhitePages.sln`.
   - **JMS sample:** `JMSBiztalkGreeter.sln`.

   This launches Visual Studio 2005 and opens the solution as shown, for example, in Figure 6 on page 29.

Figure 6. BizTalk CORBA Solution



Project File	WhitePages.btproj
Project Folder	C:\Program Files\IONA\Artix Conne

Step 4: Launching the Artix Connect for WCF Wizard

Steps

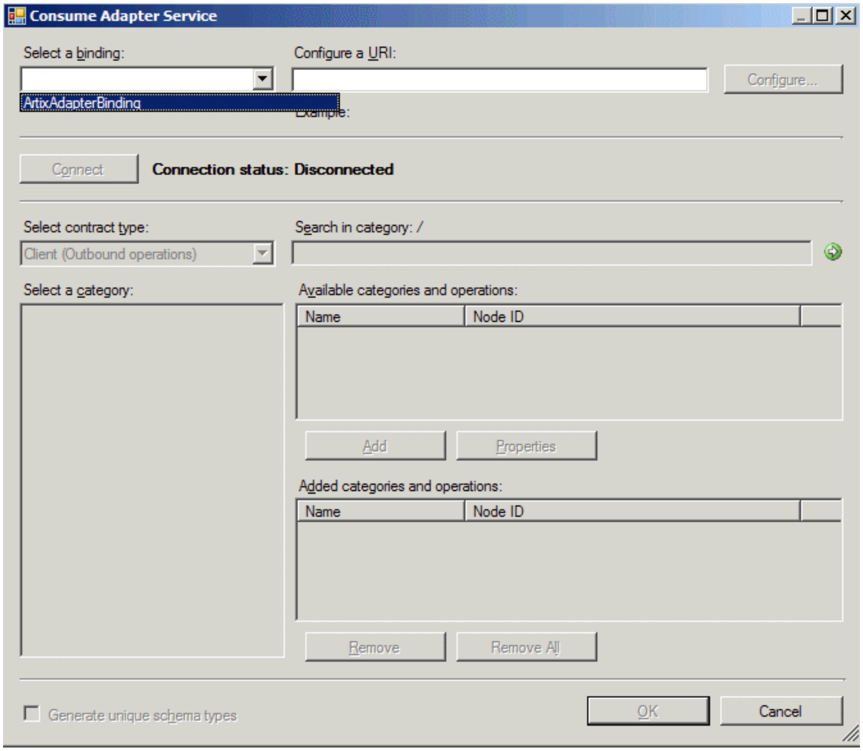
The first step in using Artix Connect for WCF is to the Microsoft LOB adapter framework.

1. In the Visual Studio Solution Explorer window, right-click on the sample application project and select **Add | Add Generated Items** from the context menu.

2. In the Add Generated Items dialog, choose **Consume Adapter Service** and click **Add**.

3. In the Consume Adapter Service dialog, from the Select a binding drop-down list, select **ArtixAdapterBinding**, as shown in Figure 7 on page 31.

Figure 7. Add Consume Adapter Service Dialog



Consume Adapter Service

Select a binding: ArtixAdapterBinding

Configure a URI:

Example:

Connection status: Disconnected

Select contract type: Client (Outbound operations)

Search in category: /

Select a category:

Available categories and operations:

Name	Node ID

Added categories and operations:

Name	Node ID

□ Generate unique schema types

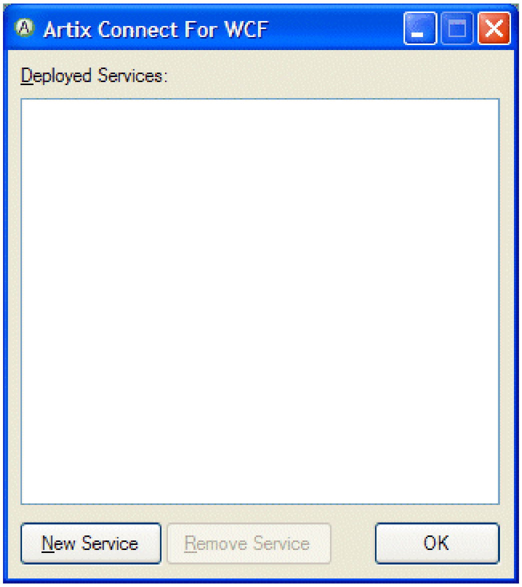
4. Click **Configure**.

5. In the Configure Adapter dialog, click **OK**.

6. In the Consume Adapter Service dialog, click **Connect**.

   The Artix Connect for WCF wizard opens as shown in Figure 8 on page 32. The deployed services list is empty if you have not already deployed any services.

Figure 8. Artix Connect for WCF Wizard



Step 5: Connecting to Target External Service

Connecting to CORBA

Steps

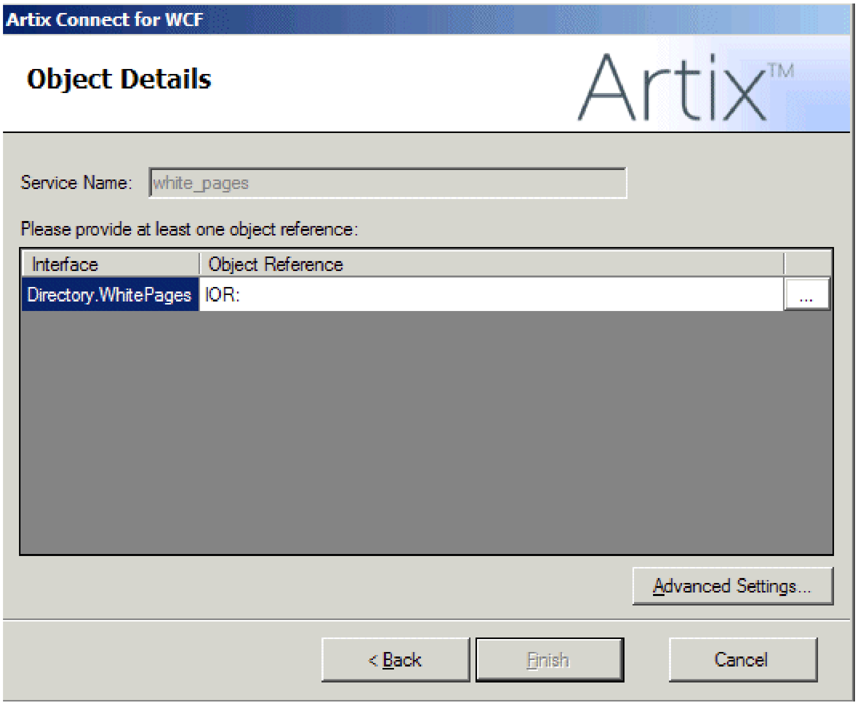
To use the Artix Connect for WCF wizard to connect to the CORBA system:

1. In the Artix Connect for WCF wizard, click **New Service**.
2. In the New Service window, select the CORBA radio button and click **OK**.
3. In the IDL File Selection window, click **Browse**.
4. In the Select IDL File dialog box, browse to the location of the sample CORBA system IDL file, which is located in the following directory of your Artix Connect for WCF installation:

   ```
   InstallDir\Visual Studio Adapter\samples\biztalk\corba\etc
   ```

5. Select the `white_pages.idl` file and click **Open**.
6. In the Select IDL File Selection window, click **Next**. The wizard checks that the IDL file is valid.
7. In the Object Details window, the interface defined in the IDL file is displayed, as shown in Figure 9 on page 35.

Figure 9. CORBA Object Details Window



Artix Connect for WCF

Object Details

Service Name: white_pages

Please provide at least one object reference:

Interface	Object Reference
Directory.WhitePages	IOR:

8. To provide the CORBA service object reference, click **...** and browse to the location of the sample CORBA system IOR file. It is located in the same directory as the IDL file; that is:

   ```
   InstallDir\Visual Studio Adapter\samples\biztalk\corba\etc
   ```

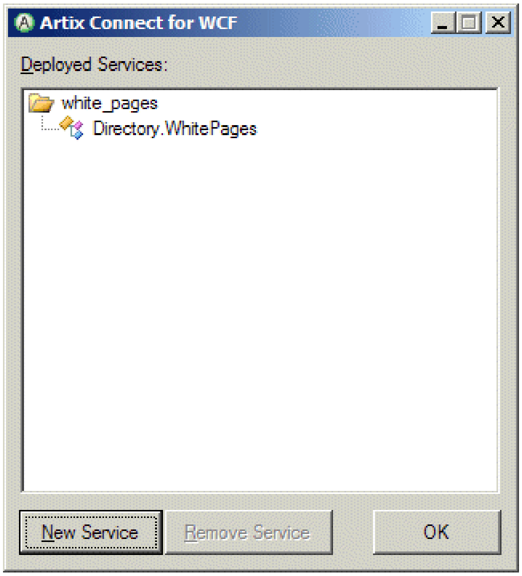
9. Select the `white_pages.ior` file and click **Open**.

   The wizard adds the IOR file to the Object Reference field of the Object Details window.

10. Click **Finish**.

    In the Artix Connect for WCF wizard, the CORBA white pages system is added to the list of deployed services (see Figure 10 on page 36).

Figure 10. CORBA White Pages Added to Deployed Services List



Connecting to JMS

Introduction

The default JMS broker used with the sample application is FUSE Message Broker. If you want to use any of the other supported JMS brokers, please refer to Using one of the other JMS brokers in the *Getting Started Guide*. It provides you with prerequisite steps that you need to complete before using another JMS broker. In addition, for each JMS broker, it includes the JMS broker and destination settings that you need when working through the steps in this section.

Note

When the instructions for using the other supported brokers point to the sample directory, they refer to the CORBA JMS sample application that is also included with Artix Connect for WCF. Follow the instructions, but where you are asked to make changes to a file in the CORBA JMS sample application, make the changes instead to the equivalent file in the BizTalk JMS sample application.

Selecting a JMS Broker

To use Artix Connect for WCF to connect the JMS system:

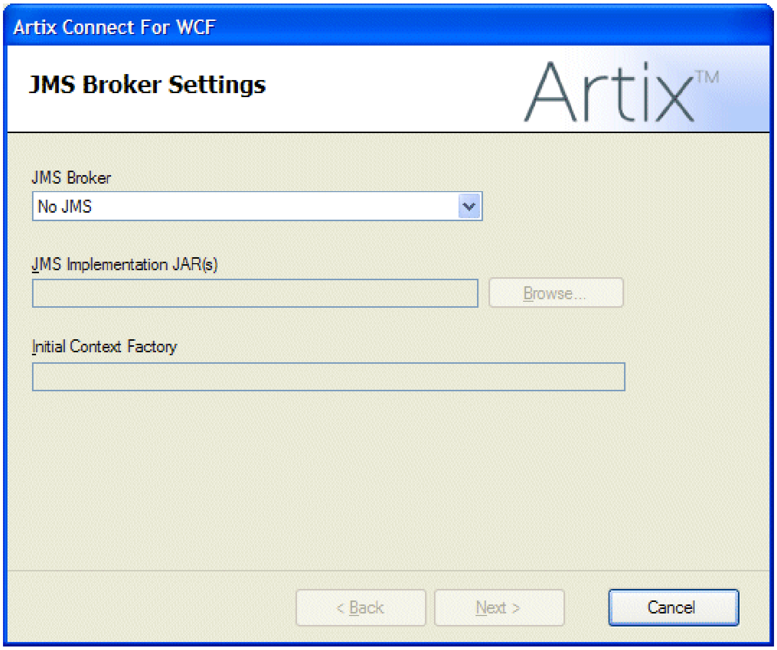
1. In the Artix Connect for WCF wizard, click **New Service**.

2. In the New Service window, select the JMS radio button.

3. Click **Next**.

4. In the JMS Broker Settings window, shown in Figure 11 on page 38.

Figure 11. Adding JMS Broker Settings



Artix Connect For WCF

JMS Broker Settings

JMS Broker: No JMS

i. Under **JMS Broker**, select **ActiveMQ or FUSE Message Broker**.

   Note that the Initial Context Factory is set automatically when you select a JMS broker.

ii. Under **JMS Implementation JAR(s)**, click **Browse** and select the implementation JAR for the FUSE Message Broker or Apache ActiveMQ version that you are using. For example, for FUSE Message Broker 5.0.0.9, browse to the top level of the FUSE Message Broker installation directory and select the `activemq-all-5.0.0.9-fuse.jar` file.

   For a complete list of JMS implementation JARs, see JMS Broker Implementation JARs in the *Installation Guide*.

iii. Click **Next**.

Note

You are only asked to set JMS broker settings once. The JMS Broker Settings window does not appear when you run the Artix Connect for WCF wizard again. If you want to subsequently change the JMS broker that you are using, please use the Artix Administration tool to enter details of the new broker. For details, see Using one of the other JMS brokers in the *Getting Started Guide*.

Selecting a payload format

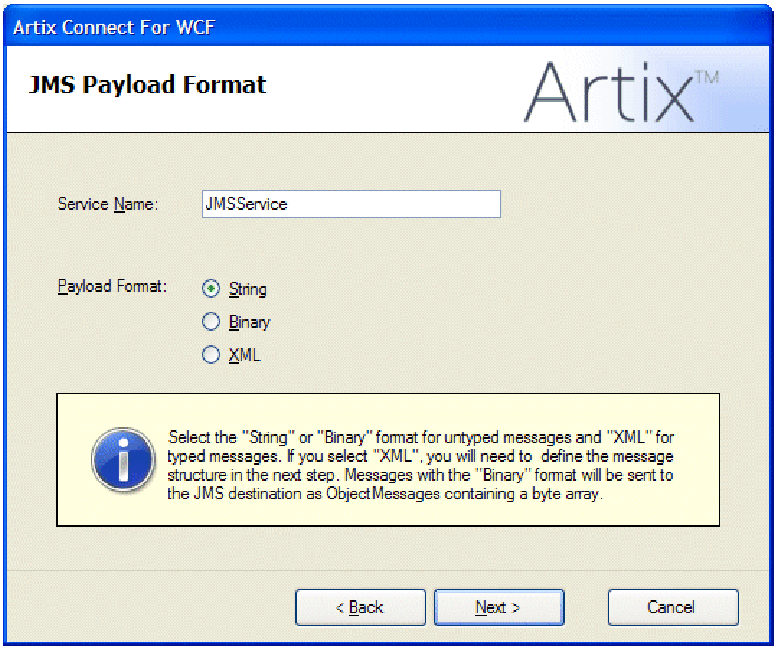
The JMS Payload Format window enables you to give the service a name and to select the type of message that you are sending. In the sample application, the message type is `String`.

To set the JMS payload format for the sample application:

1. Leave the Service Name as `JMSService`.

2. Under Payload Format, select the **String** radio button, as shown in Figure 12 on page 40.

Figure 12. Adding JMS Service Name and Payload Format Details



Service Name: JMSService

Payload Format: ● String ○ Binary ○ XML

Select the "String" or "Binary" format for untyped messages and "XML" for typed messages. If you select "XML", you will need to define the message structure in the next step. Messages with the "Binary" format will be sent to the JMS destination as ObjectMessages containing a byte array.

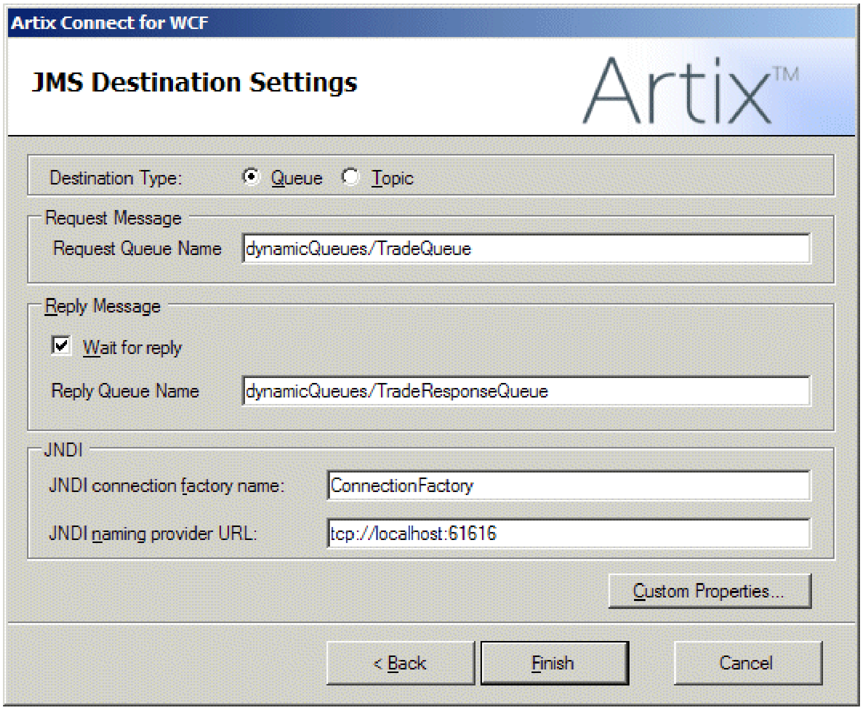
3. Click **Next**.

If your were sending an XML message, you would need to define the message structure. For the purposes of the sample application, this is not necessary. If, however, you want to see an example of defining an XML message, refer to Selecting a payload format in the *Getting Started Guide*.

Specifying JMS destination settings

In the JMS Destination Settings window you need to set JMS destination information (see Figure 13 on page 41). This information is specific to the JMS service to which you want to connect and the JMS broker that you are using.

Figure 13. JMS Destination Settings



JMS Destination Settings

Destination Type: ● Queue ○ Topic

Request Message

Request Queue Name: dynamicQueues/TradeQueue

Reply Message

☑ Wait for reply

Reply Queue Name: dynamicQueues/TradeResponseQueue

JNDI

JNDI connection factory name: ConnectionFactory

JNDI naming provider URL: tcp://localhost:61616

1. Fill in the JMS Destination Settings window. The fields are described in Table 1 on page 41. For the sample application, use the settings shown in Figure 13 on page 41.

   For example settings for the other supported JMS brokers, see the Setting up Your JMS Broker in the *Getting Started Guide*.

Table 1. JMS Destination Settings

Field	Description
Destination Type	Specifies whether you are connecting to a JMS queue or topic.
Request Queue/Topic Name	Specifies the name of the JMS queue or topic to which you are trying to connect.
Reply Queue/Topic Name	Specifies the name of the JMS queue or topic to which the reply, if there is one, is sent.

Field	**Description**
JNDI connection factory name	Specifies the name of the JMS broker connection factory.
JNDI naming provider URL	Specifies the URL used to locate and connect to the JMS broker.

2. You can set custom properties; for example, if the JMS service requires an access user name and password, as follows:

   ## Note

   This is not required for the sample application.

   i. Click **Custom Properties**.

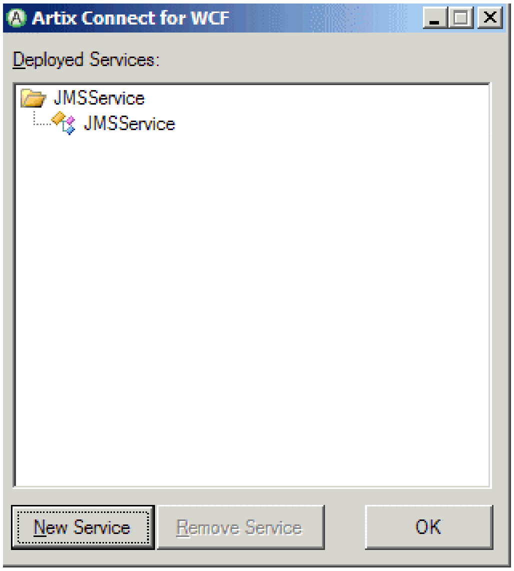
   ii. In the Custom Properties window, under **Name** type the name of your custom property, and under **Value** type the value of your custom property.

   iii. Click **OK**.

3. Click **Finish**.

   The wizard completes its tasks and the JMS service is listed under Deployed Services, as shown in Figure 14 on page 43.

Figure 14. JMS Service Deployed



4. Click **OK**.

   The Artix Connect for WCF wizard completes and returns to the Consume Adapter Service wizard.

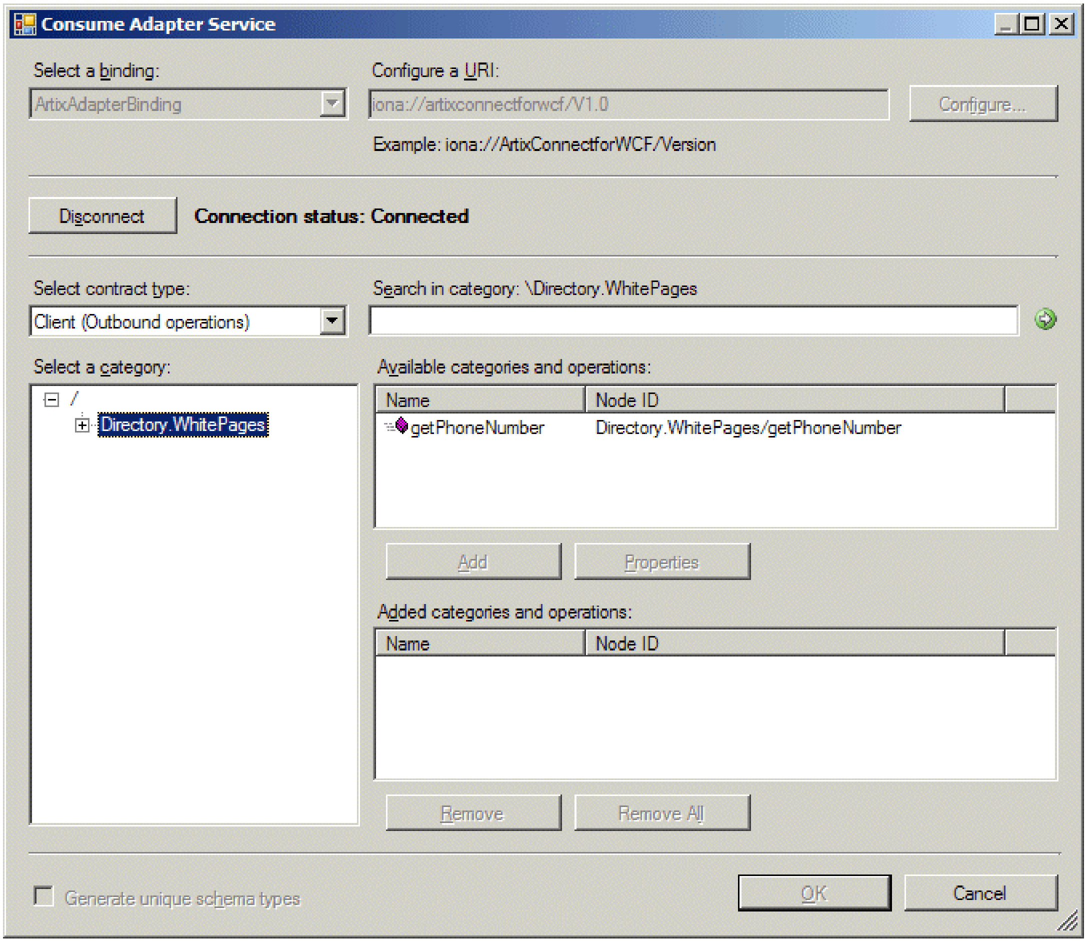
Step 6: Making CORBA and JMS Operations Available to BizTalk

Introduction

The Consume Adapter Service wizard lists the service that you just deployed in the **Select a category** panel (see, for example, Figure 15 on page 45). The **OK** button is disabled. It remains so until you specify which operations you want to use with BizTalk. The operations that you choose are made available to BizTalk through the schema files that Artix Connect for WCF generates.

Artix Connect for WCF supports both one-way and two-way message exchange patterns.

Figure 15. CORBA Operations



Select a binding: ArtixAdapterBinding

Configure a URI: iona://artixconnectforwcf/V1.0

Example: iona://ArtixConnectforWCF/Version

Connection status: Connected

Select contract type: Client (Outbound operations)

Search in category: \Directory.WhitePages

Select a category: / Directory.WhitePages

Available categories and operations:

Name	Node ID
getPhoneNumber	Directory.WhitePages/getPhoneNumber

Added categories and operations:

Name	Node ID

□ Generate unique schema types

Steps

To make the operations available to BizTalk, complete the following steps:

1. In the Consume Adapter Service wizard, under the **Select a category** panel, select the service that you just deployed:

   - **CORBA sample:** `Directory.WhitePages`

- **JMS sample:** `JMSService`

2. In the **Available categories and operations** panel, select the operations that you want to use:

   - **CORBA sample:** `getPhoneNumber`
   - **JMS sample:** `jmsRequest`

3. Click **Add** to add the operations to the **Added categories and operations** panel.

4. Click **OK**.

The wizard starts to generate the XML schema definitions and binding metadata required by BizTalk to enable it to communicate with the target external service.

Step 7: Deploying your BizTalk Project

Introduction

Once you have used Artix Connect for WCF to generate the XML schema definitions and binding metadata needed to enable BizTalk to communicate with the CORBA service or the JMS service, you must configure and deploy receive and send locations. The generated `.dll` needs to be deployed into the runtime environment to be accessible to BizTalk.

Note

For the purposes of the sample applications, configuration is done using the Visual Studio orchestration file. You can also configure the send and receive locations using the BizTalk Administration Console. See your BizTalk documentation for details.

Receive and Send locations

In both the CORBA and the JMS sample applications, the receive location is a folder on your machine to which BizTalk has read access. Any messages saved to this folder will be published to BizTalk. The send location is a file on your machine to which BizTalk publishes any reply messages.

Configuring a receive location

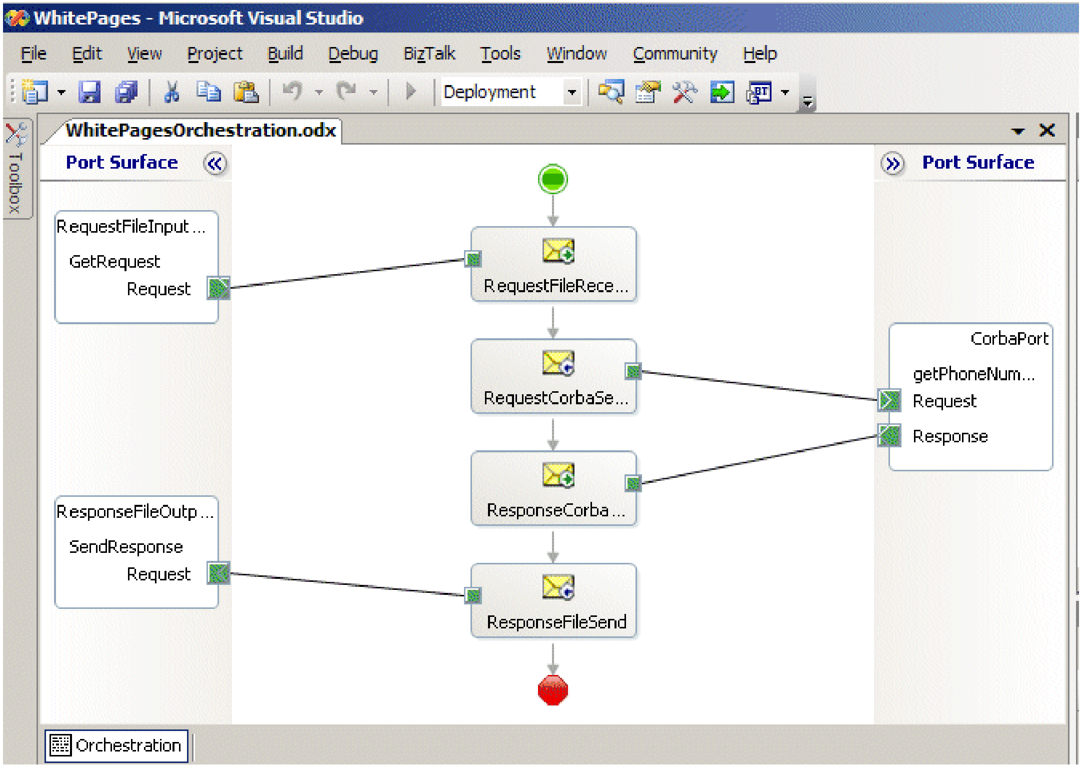
To configure a receive location:

1. In the Visual Studio Solution Explorer, double click on the sample `.odx` file:

   - **CORBA sample:** `WhitePagesOrchestration.odx`.
   - **JMS sample:** `GreeterOrchestration.odx`.

   The orchestration appears, for example, as shown in Figure 16 on page 48.

Figure 16. BizTalk CORBA Orchestration

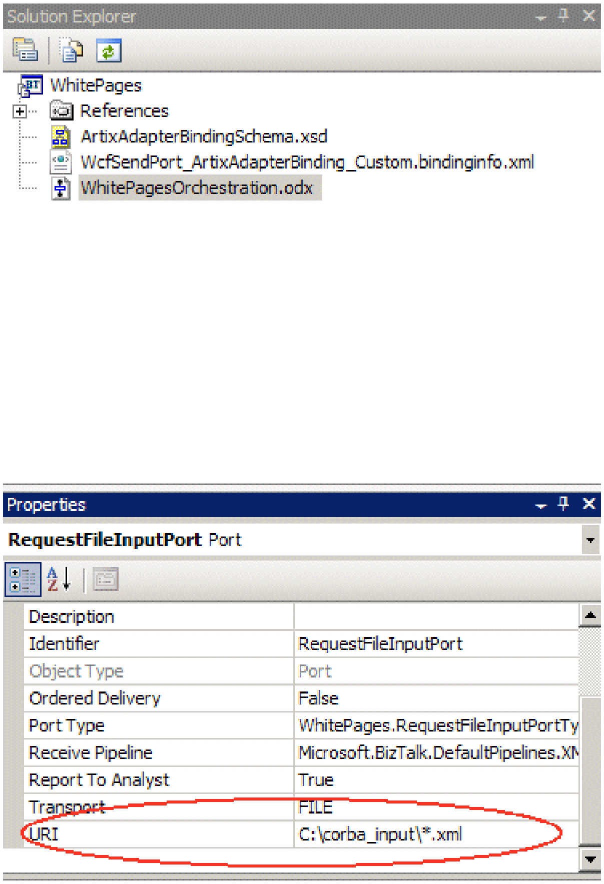


2. In the Orchestration Designer, right-click on:
   - **CORBA sample:** `RequestFileInputPort`.
   - **JMS sample:** `IncomingMessagePort`

   The port is represented in the top-left corner of the orchestration (the name may appear truncated).
3. Select **Properties Window** from the context window.
4. In the **Properties** window, set the URI property to *`MyFolder`*`\*.xml`, where *`MyFolder`* is a clean folder on your file system to which BizTalk has read

access; for example, `C:\corba_input\*.xml` as shown in Figure 17 on page 49.

Figure 17. Configuring a Receive Location



Solution Explorer

- WhitePages
  - References
  - ArtixAdapterBindingSchema.xsd
  - WcfSendPort_ArtixAdapterBinding_Custom.bindinginfo.xml
  - WhitePagesOrchestration.odx

Properties

RequestFileInputPort Port

Description	
Identifier	RequestFileInputPort
Object Type	Port
Ordered Delivery	False
Port Type	WhitePages.RequestFileInputPortTy
Receive Pipeline	Microsoft.BizTalk.DefaultPipelines.XM
Report To Analyst	True
Transport	FILE
URI	C:\corba_input\*.xml

Configuring a send location

To configure a send location:

1. In the Orchestration Designer, right-click on:

   - **CORBA sample:** `ResponseFileOutputPort`.

- **JMS sample:** `OutgoingMessagePort`.

  The port is represented in the bottom-left corner of the orchestration (the name may appear truncated).

2. Select **Properties Window** from the context window.
3. Set the URI property to a file to which the output messages can be written. The file does not need to exist on your file system, but the location does; for example, `C:\output.xml`.

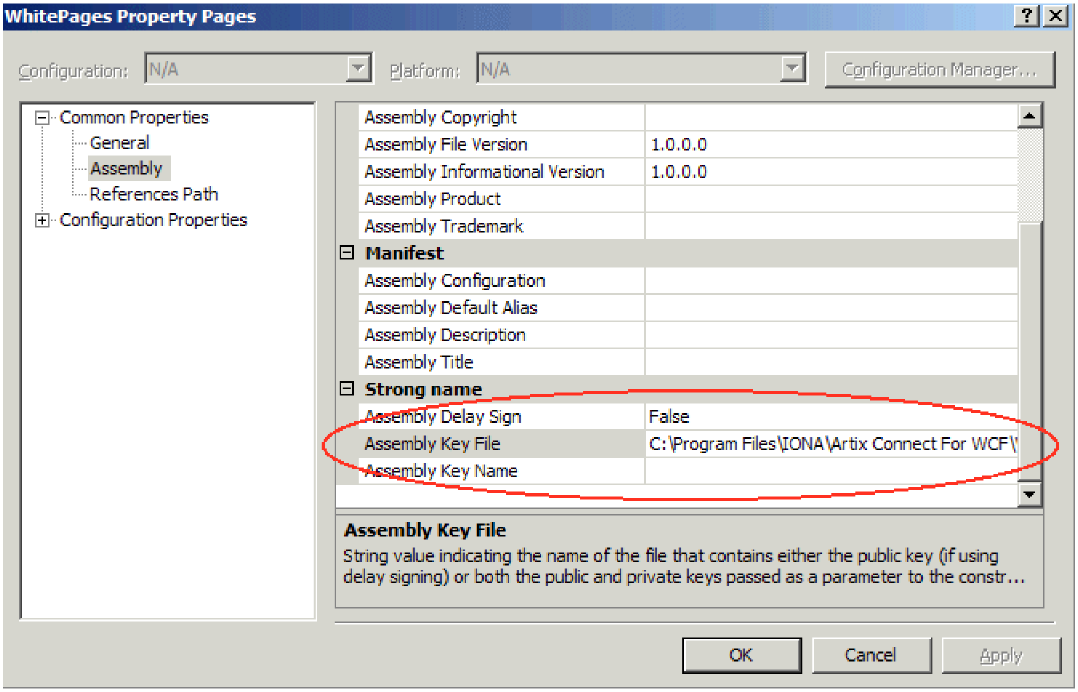
Assigning a strong name to your assembly

Before you can deploy your BizTalk project, you must assign a strong name to the assembly. A strong-named assembly provides security benefits and guarantees the uniqueness of the assembly by assigning a digital signature and a unique key pair.

To assign the sample assembly with a strong name, complete the following steps:

1. In the Visual Studio Solution Explorer, right-click on the project.
2. Select **Properties** from the context menu.
3. In the Property Pages dialog box, click **Assembly**.
4. In the right pane, scroll down to the Strong name section.
5. Click the field to the right of Assembly Key File, and click the ellipsis (…) as shown in Figure 18 on page 51.

Figure 18. Assigning a Strong Name



Configuration: N/A

Platform: N/A

Assembly Copyright	
Assembly File Version	1.0.0.0
Assembly Informational Version	1.0.0.0
Assembly Product	
Assembly Trademark	
Manifest	
Assembly Configuration	
Assembly Default Alias	
Assembly Description	
Assembly Title	
Strong name	
Assembly Delay Sign	False
Assembly Key File	C:\Program Files\IONA\Artix Connect For WCF\
Assembly Key Name	

Assembly Key File

String value indicating the name of the file that contains either the public key (if using delay signing) or both the public and private keys passed as a parameter to the constr...

6. In the Assembly Key File dialog box, select:

- **CORBA sample:** `WhitePages.snk` file by navigating to the following directory of your Artix Connect for WCF installation:

  ```
  InstallDir\Visual Studio
  Adapter\samples\biztalk\corba\biztalk
  ```

- **JMS sample:** `JmsBiztalkGreeter.snk` file by navigating to the following directory of your Artix Connect for WCF installation:

  ```
  InstallDir\Visual Studio
  Adapter\samples\biztalk\jms\biztalk
  ```

7. Click **Open**.

8. Click **OK**.

Deploying your solution

To deploy the sample solution:

From the **Build** menu, select **Deploy** ***`SampleApplication`*****.**.

Step 8: Configuring BizTalk Send and Receive Ports

Introduction

In this section, you use the BizTalk Administration Explorer to finalize the port configuration.

Configuring the receive port

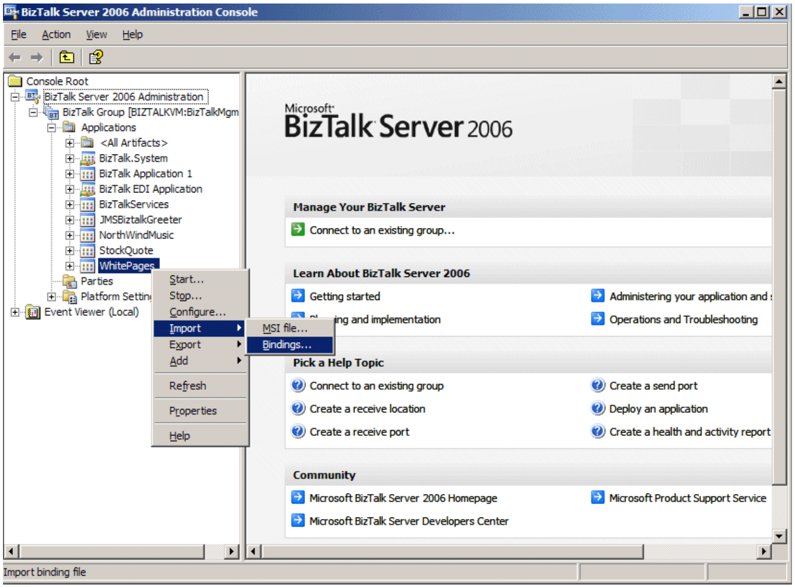
To configure the receive port:

1. Open the BizTalk Administration Console from the Windows Start menu as follows:

   (All) Programs | Microsoft BizTalk Server 2006 | BizTalk Server Administration

2. Expand the BizTalk Server Administration node.
3. Expand the BizTalk Group node.
4. Expand the Applications node.
5. Right-click on the newly deployed application:
   - **CORBA sample:** `WhitePages`.
   - **JMS sample:** `JMSBiztalkGreeter`.
6. Choose **Import | Bindings**, as shown, for example, in Figure 19 on page 54.

Figure 19. Importing Bindings


BizTalk Server 2006 Administration Console
File
Action
View
Help
Console Root
BizTalk Server 2006 Administration
BizTalk Group [BIZTALKVM:BizTalkMgm
Applications
<All Artifacts>
BizTalk.System
BizTalk Application 1
BizTalk EDI Application
BizTalkServices
JMSBiztalkGreeter
NorthWindMusic
StockQuote
WhitePages
Parties
Platform Settin
Event Viewer (Local)
Start...
Stop...
Configure...
Import
Export
Add
Refresh
Properties
Help
MSI file...
Bindings...
Microsoft
BizTalk Server 2006
Manage Your BizTalk Server
Connect to an existing group...
Learn About BizTalk Server 2006
Getting started
Administering your application and
Pl...ing and implementation
Operations and Troubleshooting
Pick a Help Topic
Connect to an existing group
Create a send port
Create a receive location
Deploy an application
Create a receive port
Create a health and activity report
Community
Microsoft BizTalk Server 2006 Homepage
Microsoft Product Support Service
Microsoft BizTalk Server Developers Center
Import binding file

7. In the Import Bindings dialog, navigate to `WcfSendPort_ArtixAdapterBinding_Custom.bindinginfo.xml`, which is located in the sample application Visual Studio project at:

   - **CORBA sample:** `*InstallDir*\Visual Studio Adapter\samples\biztalk\corba\biztalk\WhitePages`

   - **JMS sample:** `*InstallDir*\Visual Studio Adapter\samples\biztalk\jms\biztalk\JMS Biztalk Greeter`

8. Click **Open**.

Configuring the send port

To configure the send port:

1. Right-click on the application again.

2. Click **Configure**.

3. In the Configure Application dialog:

   - **CORBA sample:**

     i. Select **WhitePagesOrchestration**.

     ii. In the Host drop-down list, select the `BizTalkServerApplication`.

     iii. Under **Outbound Logical Ports**, select `CorbaPort`.

     iv. From the Send Ports/Send Port Groups drop-down list, select `WcfSendPort_ArtixAdapterBinding_Directory.WhitePages_Custom`.

     v. Click **Apply**.

     vi. Click **OK**.

   - **JMS sample:**

     i. Select **BizTalk_Orchestration1**.

     ii. In the Host drop-down list, select the `BizTalkServerApplication`.

     iii. Under **Outbound Logical Ports**, select `JMSBackendPort`.

     iv. From the Send Ports/Send Port Groups drop-down list, select `WcfSendPort_ArtixAdapterBinding_JMSService_Custom`.

     v. Click **Apply**.

     vi. Click **OK**.

Configuring send port FILE transport

The BizTalk sample applications require that more than one message can be sent. To ensure that this is possible, you must configure the send port `FILE` transport to allow the files to be overwritten. To set the `FILE` transport properties to overwrite:

1. Expand the deployed solution node:

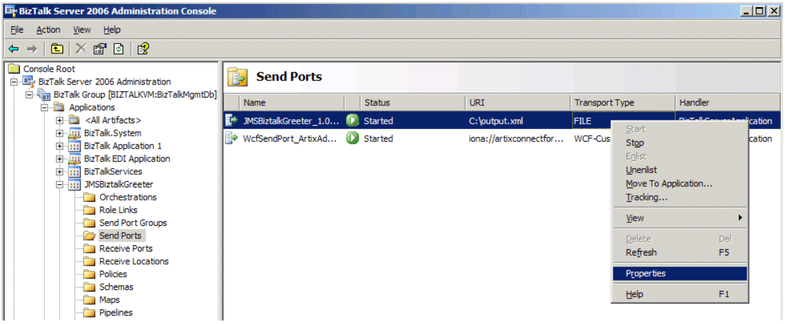
   - **CORBA sample:** `WhitePages`

   - **JMS sample:** `JMSBiztalkGreeter`.

2. Select the **Send Ports** folder.

3. Under Transport Type, right-click on **FILE** and click **Properties**, as shown in Figure 20 on page 56.

Figure 20. Setting File Transport Properties

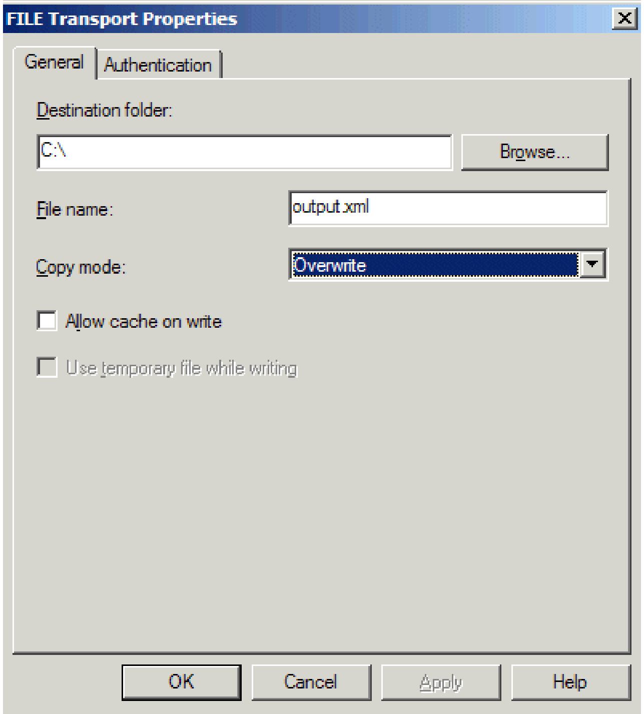


Name	Status	URI	Transport Type	Handler
JMSBiztalkGreeter_1.0...	Started	C:\output.xml	FILE	BizTalkServerApplication
WcfSendPort_ArtixAd...	Started	iona://artixconnectfor...	WCF-Cus	cation

4. In the properties dialog, click **Configure**.

5. In the FILE Transport Properties dialog, shown in Figure 21 on page 57, from the **Copy mode** drop-down list, select Overwrite.

Figure 21. Setting FILE Transport Copy Mode



Destination folder: C:\

File name: output.xml

Copy mode: Overwrite

☐ Allow cache on write

☐ Use temporary file while writing

6. Click **Apply** followed by **OK**.

7. In the properties dialog, click **Apply** followed by **OK**.

Step 9: Running the Sample Application

Running the Sample Application

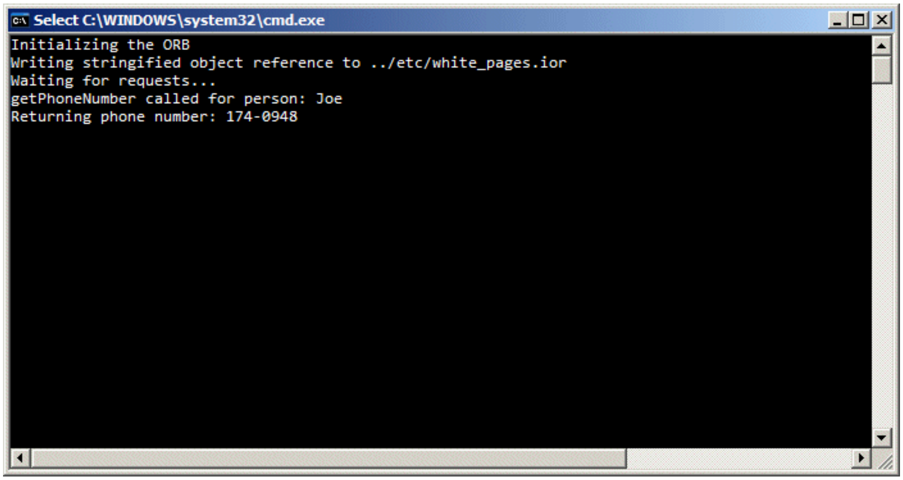
To run the sample application:

1. In the BizTalk Server 2006 Administration Console, right-click on the sample application and click **Start**.
2. In the Start Application dialog, click **Start**.
3. Return to the Visual Studio project.
4. Right-click on `ArtixAdapterBindingSchema.xsd` file and select **Generate Instance** from the context menu.

   An XML file is generated at the location shown in the Output window (you may have to scroll to the right to see it).
5. Open the XML file in notepad and replace the contents of:
   - **CORBA sample:** `<person>` element with any name; for example, Joe.
   - **JMS sample:** `<ns0:requestContentElement>` element with any name.
6. Save this file to the receive location that you defined in Configuring a receive location on page 47.

   After a few moments, you will see the message has been received by the CORBA server if your running the CORBA sample or the Java server if you are running the JMS sample. See, for example, Figure 22 on page 59.

Figure 22. Message Received by CORBA Server



```
Initializing the ORB
Writing stringified object reference to ../etc/white_pages.ior
Waiting for requests...
getPhoneNumber called for person: Joe
Returning phone number: 174-0948
```

Subsequently, the file that you configured as part of the send port in Configuring a send location on page 49; that is, `C:\output.xml`, should appear and contain the response message.

Index