

# **Microsoft BizTalk Server 2006 Part-II**

## Table of Contents

### [Module 1: Business Solutions Scenarios](#)

### [Module 2: Service Oriented Solution](#)

Understanding the Service Oriented Solution

### [Module 3: Business Process Management Solution](#)

### [Module 4: Patterns in the Business Process Management Solution](#)

Designing with Patterns: the Business Process Management Solution with Patterns

Translating the Patterns of the Business Process Management Solution

Pattern Catalog for the Business Process Management Solution

Components of the Business Process Management Solution

### [Module 5: Processing in the Business Process Management Solution](#)

Processing in the OrderBroker Orchestration

### [Module 6: Process Manager Logic](#)

Interrupt Handling in the Business Process Management Solution

### [Module 7: Exception Handling in the Business Process Management Solution](#)

The ExceptionHandler Orchestration

Custom Exceptions

### [Module 8: Implementation Highlights of the Business Process Management Solution](#)

Validation with Business Rules

Number of Processing Stages

Inverse Direct Binding

Publishing to the MessageBox Database

Using SSO Efficiently in the Business Process Management Solution

Using the Order Handler Object to Communicate with Backend Systems

Merging XML Documents

### [Module 9: Merging XML Documents](#)

Using the Ops Adapter

The Sample Operations Error Handler

Ops Adapter Implementation Details

The Recaller Object

Monitoring the Business Process Management Solution with BAM

Building Business Process Management Solutions

Versioning the Business Process Management Solution

### [Module 10: Business Process Management Solution Reference](#)

File Inventory for the Business Process Management Solution

Message Reference for the Business Process Management Solution

### [Module 11: Deploying the Business Process Management Solution](#)

### [Module 12: Developer Machine Setup for the BPM Solution](#)

Before Installing the Business Process Management Solution

How to Install the Business Process Management Solution

How to Run the Business Process Management Solution

### **Module 13: Deploying the BPM Solution to Production Servers**

- Prerequisites of the BPM Solution Production Servers
- Preparing the BPM Solution to be Deployed
- Preparing the Production Servers for the BPM Solution
- Installing the BPM Solution to Production Servers
- Running the BPM Solution on Production Servers

### **Module 14: Business-to-Business Solution**

### **Module 15: Developing a Business-to-Business Solution**

### **Module 16: Patterns in the Business-to-Business Solution**

- Designing with Patterns: the Business-to-Business Solution
- Translating the Patterns of the Business-to-Business Solution
- Pattern Catalog for the Business-to-Business Solution
- Components of the Business-to-Business Solution

### **Module 17: Processing in the Business-to-Business Solution**

- Processing Orders in the Business-to-Business Solution
- Getting Profile Information in the Business-to-Business Solution
- Retrieving Orders in the Business-to-Business Solution
- Updating Orders in the Business-to-Business Solution
- Retrieving Partner Profiles in the Business-to-Business Solution
- Updating Partner Profiles in the Business-to-Business Solution
- Using EDI in the Business-to-Business Solution

### **Module 18: Implementation Highlights of the Business-to-Business Solution**

- Trading Partner Management in the Business-to-Business Solution
- Message Brokering in the Business-to-Business Solution
- Certificates and Signing Messages

### **Module 19: Business-to-Business Solution Reference**

- File Inventory for the Business-to-Business Solution
- Message Reference for the Business-to-Business Solution

### **Module 20: Developer Machine Setup for the Business-to-Business Solution**

- Before Installing the Business-to-Business Solution
- How to Install the Business-to-Business Solution
- How to Run the Business-to-Business Solution

### **Module 21: Deploying the Business-to-Business Solution to Production Servers**

- Prerequisites of the B2B Solution Production Servers
- How to Prepare the B2B Solution for Deployment
- Preparing the Production Servers for the B2B Solution
- Installing the B2B Solution to Production Servers
- Running the B2B Solution on Production Servers
- How BizTalk Server Processes Large Messages

### **Module 22: Development**

### **Module 23: Developing BizTalk Server Applications**

**Module 24: Developer Tools**

About BizTalk Project System  
Using the BizTalk Project System

**Module 25: Using Visual Studio**

Using the File Menu  
Using the View Menu  
Using the Project Menu  
Using the Build Menu  
Using the Debug Menu  
Using the BizTalk Menu  
Using the Help Menu  
Using Property Pages

**Module 26: Working with BizTalk Projects**

BizTalk Server Project Templates

**Module 27: Creating BizTalk Projects**

Considerations When Creating BizTalk Projects  
Adding Project Items

**Module 28: Configuring Project Property Pages**

Common Properties: General  
Common Properties: Assembly  
Common Properties: References Path

**Module 29: Configuration Properties**

Configuration Properties: Build  
Configuration Properties: Deployment  
Solution Build Configurations  
Source Control and the BizTalk Project System

**Module 30: Using BizTalk Type Browser**

BizTalk Type Browser Options

**Module 31: BizTalk Type Browser Nodes**

Root and Container Nodes  
Project/Assembly or Reference Nodes  
Type Nodes

**Module 32: About BizTalk Namespace References Included in BizTalk Projects**

Microsoft.BizTalk.DefaultPipelines Reference  
Microsoft.BizTalk.GlobalPropertySchemas Reference  
System Reference  
System.Xml Reference

**Module 33: Using the BizTalk Messaging Engine**

Interchange Processing  
Ordered Delivery of Messages

**Module 34: Error Handling**

Using Failed Message Routing  
Using Acknowledgments

**Module 35: Creating Schemas Using BizTalk Editor**

Planning for Schema Creation

**Module 36: About Instance Messages**

Structure of an XML Message

Structure of a Flat File Message

**Module 37: Structure of an XML Message**

XML Message Envelopes

Nested XML Message Envelopes

**Module 38: Structure of a Flat File Message**

Flat File Message Headers

Flat File Message Bodies

Flat File Message Trailers

Flat File Messages with Positional Records

Flat File Messages with Delimited Records

Migrating Flat File Records

**Module 39: About Schemas**

Different Types of BizTalk Schemas

Role of XSD

XSD Resources on the Web

BizTalk Representation of Schemas

Schemas That Use Other Schemas

Type Reuse and Derivations

Schema Migration from Previous Versions of BizTalk Server

Ways to Use Message Content to Control Message Processing

Schema Validation in Visual Studio

Instance Message Generation and Validation

**Module 40: Different Types of BizTalk Schemas**

XML Schemas

Flat File Schemas

Envelope Schemas

Property Schemas

**Module 41: BizTalk Representation of Schemas**

Nodes That Correspond Directly to Message Instance Data and Structure

Nodes That Control Instance Message Structure and Variations

Nodes That Simplify Schema Creation

Nodes That Provide Wildcard Capabilities

Node Properties

**Module 42: Nodes That Correspond Directly to Message Instance Data and Structure**

Root Nodes

Record Nodes

Field Element Nodes

Field Attribute Nodes

**Module 43: Nodes That Control Instance Message Structure and Variations**

- Sequence Group Nodes
- Choice Group Nodes
- All Group Nodes
- <Equivalent> Nodes and Their Child Nodes

**Module 44: Nodes That Simplify Schema Creation**

- Schema Node
- Attribute Group Nodes

**Module 45: Nodes That Provide Wildcard Capabilities**

- Any Element Nodes
- Any Attribute Nodes

**Module 46: Node Properties**

- Node Name Property
- Property Interdependencies
- Additional Flat File Properties

**Module 47: Node Name Property**

- Which Node Name Characters Get Encoded
- How Node Name Characters Get Encoded

**Module 48: Type Reuse and Derivations**

- Complex Global Type Definition and Naming
- Ways to Use Complex Global Types
- Simple type derivations

## Business Solutions Scenarios

This section describes four complete, model BizTalk applications. Each application represents a particular business pattern and demonstrates other constituent integration patterns.

Each section presents an overview of the application, a developer's guide that describes the patterns and design decisions reached in the course of developing the application, and deployment information for building and running the application.

### In This Section

- Service Oriented Solution
- Business Process Management Solution
- Business-to-Business Solution

## Service Oriented Solution

The Service Oriented solution shows how to present a BizTalk application as a service available as a web service and in forms accessible to legacy applications. The solution also shows how to communicate with backend processes as web services as well as aggregating responses from multiple backend systems.

The sections provide an overview of the solution, detailed explanations of the patterns and design choices, information about building and running the solution.

### In This Section

- Understanding the Service Oriented Solution
- Deploying the Service Oriented Solution
- Developing a Service Oriented Solution

## Understanding the Service Oriented Solution

The service oriented solution presents a credit balance reporting application designed as a service. The application, in turn, uses three backend applications, exposed as services themselves, to get the information needed for the credit balance.

A Service Oriented Architecture (SOA) is an approach that partially overlaps building distributed systems. A service-oriented approach has several characteristics:

Loosely coupled. The application's business logic is separate from the logic of handling the service.

Discoverable. There should be a mechanism for applications to find the service.

Contractual. The interface to the service implements the contract between users and the service.

Although the literature often treats service oriented approaches as synonymous with web services, they are not. Web services present an attractive way to implement service oriented solutions, but you can use other technologies, such as .NET remoting, to create services.

For more information about service oriented architectures, see "Service Oriented Interface" at <http://go.microsoft.com/fwlink/?LinkId=46185> and "Service-Oriented Integration" at <http://go.microsoft.com/fwlink/?LinkId=46186>.

## **Reader Guidance**

The documentation for this solution assumes that you are familiar with BizTalk Server 2006 and Microsoft® Visual Studio® .NET. It also assumes that you understand basic concepts about enterprise application integration and web services.

In addition, to read and follow the developer documentation, you should be familiar with how to build applications by using Visual Studio .NET and with performing the following tasks: creating projects, setting references, and debugging and testing BizTalk solutions.

## **Credit Card Reporting at Woodgrove Bank**

The service oriented architecture solution is a credit card balance reporting service for Woodgrove Bank. Although the bank is fictional, the scenario is not—the scenario is based on an actual, deployed, customer application.

In the scenario, requests for credit card balances come from two sources:

- An Interactive Voice Response (IVR) application.

- An interactive client such as a web page or custom client application.

The solution receives requests from the IVR application through MQSeries. It handles requests from the interactive client through a web service using HTTP and SOAP.

The Woodgrove Bank application uses data from three backend, legacy systems to respond to requests:

- An application that provides the overall credit limit. This is a SAP system on a mainframe computer.

- A pending transactions system that reports the total amount for transactions pending against the account. This system is a mainframe or AS/400 system. The solution uses a web service and HIS to communicate with the mainframe.

- A payment tracking system that gives the last payment made to the system. The payment tracking system can be reached using MQSeries.

After gathering and compiling the information from the legacy systems, the solution sends the response back to the originating application and, thus, to the customer. The following block diagram shows the solution.



## Business Requirements

Because the credit reporting application responds in real time to customer requests, it must have low latency in order to handle requests quickly. In addition, it must also be able to handle high numbers of simultaneous requests. The solution uses sensitive information and a public interface so that security is significant concern. Finally, the service needs to be reliable.

For information about how the solution meets these requirements, see *Developing a Service Oriented Solution* .

## Performance Characteristics

To meet the business requirements, the scenario has the following performance characteristics:

Sustained throughput of 40 incoming requests per second.

Peak throughput of 100 incoming requests per second.

90 percent of the requests to be serviced (in and out of BizTalk Server) in under 1000 milliseconds.

95 percent of the requests to be serviced (in and out of BizTalk Server) in under 2000 milliseconds.

100 percent of the requests to be serviced (in and out of BizTalk Server) in under 5000 milliseconds.

## Three Versions of the Solution

There are three versions of the solution:

The stub version replaces all of the backend systems with software stubs. The stubs simulate the backend systems. This version provides a quick way to deploy and run the solution on a single computer.

The adapter version uses BizTalk adapters to connect to the backend systems. This version is how one might first think to implement the solution. However, the adapters introduce significant latency into getting responses back.

The inline version replaces the adapters with code that replaces the adapters. The inline version of the solution has the lowest latency and the highest throughput.

The deployment guide provides directions for building and deploying all three versions of the solution, as well as providing a way, in each version, to simulate the connection through HIS to the pending transactions system. For information about building and deploying the solution, see **Deploying a Service Oriented Solution**.

## Developing a Service Oriented Solution

The service oriented solution demonstrates a credit account balance system for Woodgrove Bank. Information about an account comes from three legacy systems: an SAP system that provides the credit limit, a pending transactions system running on a mainframe, and a payment tracking system using MQSeries. Balance check requests come through a Web service or an Interactive Voice Response (IVR) system. For more information about the scenario, see Understanding the Service Oriented Solution .

This Developer's Guide presents one approach to building a service oriented architecture solution for the Woodgrove Bank scenario. The Guide begins by considering a possible solution in terms of industry-standard patterns. Applying Patterns in the Service Oriented Solution begins the translation of that pattern into the larger structures of a BizTalk application. The next section, Components of the Service Oriented Solution, summarizes the various parts of the application and their relationships. The Implementation Details section discusses areas—such as using Enterprise Single Sign-On—that required special work to meet the criteria of the scenario. Monitoring the Service Oriented Solution with BAM describes how the solution uses the BAM API to track progress of requests and results. The Versioning the Service Oriented Solution section and the Scalling the Service Oriented Solution suggest ways of extending or modifying the solution. The reference section lists the files in the solution and provides a reference to the messages.

### In This Section

- Designing with Patterns: the Service Oriented Solution
- Translating the Patterns of the Service Oriented Solution
- Components of the Service Oriented Solution
- Implementation Highlights of the Service Oriented Solution
- Monitoring the Service Oriented Solution with BAM
- Versioning the Service Oriented Solution
- Scaling the Service Oriented Solution
- File Inventory for the Service Oriented Solution

## Patterns in the Service Oriented Solution

This section looks at the basic programming patterns making up the Service Oriented solution. It begins by considering the enterprise patterns involved in the solution, develops those patterns into BizTalk patterns. The final section catalogs the patterns in the solution, whether those are enterprise patterns, BizTalk patterns, or other general programming patterns.

### In This Section

- Designing with Patterns: the Service Oriented Solution
- Translating the Patterns of the Service Oriented Solution
- Pattern Catalog for the Service Oriented Solution

## Designing with Patterns: the Service Oriented Solution

The service-oriented solution shows how to expose a BizTalk application as a service for use by other applications. Presenting an application as a service enables other applications to easily consume the information and use it in the services that they provide. For more information about service interfaces see "Service Interface" at <http://go.microsoft.com/fwlink/?LinkId=46185>. For more information about service-oriented integration see "Service-Oriented Integration" at <http://go.microsoft.com/fwlink/?LinkId=46186>.

The solution is a credit information application that provides the information as a Web service response, after aggregating relevant information from three other applications. The application consolidates the results and returns a single message containing the summarized credit information. The three back-end systems are as follows:

**SAP Enterprise System.** The SAP back end provides the customer's overall credit limit. The solution communicates with this backend system using the SAP .NET connector.

**Pending Transactions System.** The Pending Transactions system reports the total amount of transactions outstanding against the account. The solution uses HIS to expose this system as a Web service which the BizTalk orchestration consumes.

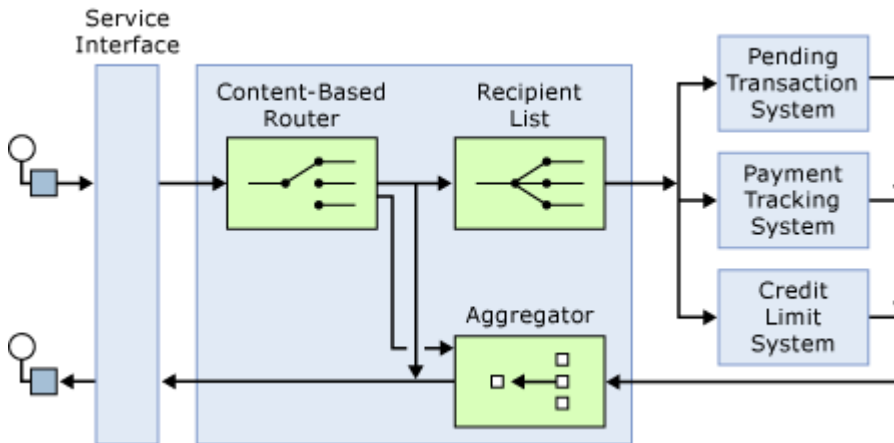
**Payment Tracking System.** The Payment Tracking system reports the last payment the individual made. This system uses MQSeries.

As you may recall from the overview of the solution, you can also use a non-Web service interface through MQSeries queues. (For more information about the general structure of the application, see Understanding the Service Oriented Solution ). Although Web services are the most common way to construct service oriented architectures, not all applications can use them. With BizTalk Server solutions you can provide, along with Web services, alternate ways for legacy applications to use the service.

The MQSeries access simulates how a legacy interactive voice response system might use the solution. The MQSeries access, along with the Web service access, shows how a single solution can be used by both legacy applications and new applications.

## Patterns Used in the Service Oriented Solution

The following diagram shows a simplified version of the patterns in the service-oriented solution.



The solution consists of four main parts, each of which represents a pattern: the service interface, a content-based router, a recipient list, and an aggregator. The service interface represents the interface mechanism that makes it possible to connect to the solution. The content-based router checks the validity of the message and sends an error message if it is invalid. The recipient list sends the message to the three back-end applications. As the back-end applications respond, the aggregator combines the responses into a single response message. The response message goes back to the requestor through the service interface.

Note that a lot is left unspecified in the diagram:

The diagram omits message translators, which are required by the solution in order to communicate with the external systems.

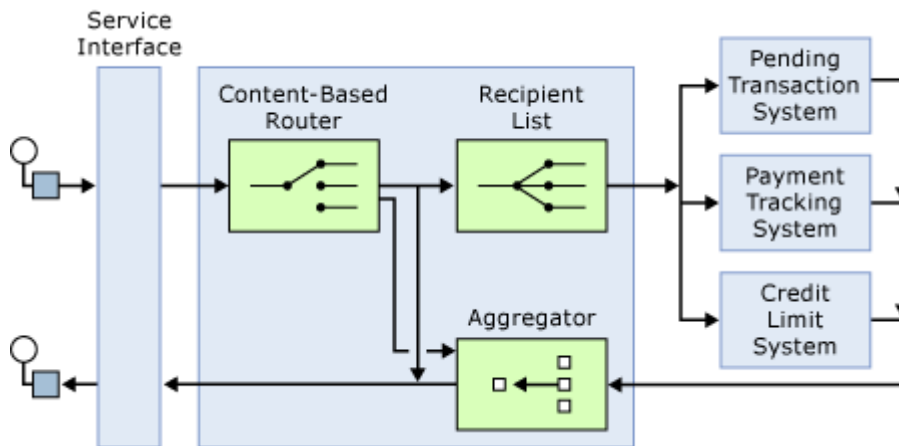
The diagram doesn't specify how to communicate with the back-end processes.

The diagram also does not specify the nature of the service interface.

Nor does the diagram indicate whether or not to use synchronous or asynchronous communication.

## Translating the Patterns of the Service Oriented Solution

This section describes how the solution translates the pattern diagram into BizTalk Server artifacts.



### Connections and Completeness Conditions

We can pretty much start anywhere with translating the patterns into BizTalk Server components. However, the connections among the components are, in effect, the infrastructure for the components so that it seems a good place to start. Also, in the case of the aggregator pattern, we need to think about what will tell it that it has all of the information it needs. This affects both its connections to other components as well as its design.

The solution needs to provide a convenient and consistent way of being used. The service interface specifies how other applications can use it. Because the solution needs to communicate with a legacy application using IBM WebSphere MQ, IBM WebSphere MQ needs to be part of the service interface. However, for newer applications, a Web service interface seems an obvious choice. A Web service interface provides maximum flexibility for other applications using the service. Here, we can use the flexibility of BizTalk Server to have a dual service interface. For information about exposing orchestrations as Web services, see [How to Map Orchestrations to Web Services](#).

The other connections are those between the service interface and the recipient list, between the recipient list and the back-end applications, and among the back-end applications, aggregator, and the service interface.

If the connection to the recipient list is synchronous, the solution does not need to use correlations to ensure all messages to the recipient list are sent and received. The connections between the back-end applications and the aggregator can be synchronous for the same reasons. The aggregator the responses from all three of the back-end applications in order to complete the query response. needs all three responses. Response times should be short so that synchronous connections are appropriate and simplify the solution.

In the aggregator pattern, there is usually a completeness condition. In cases where there are multiple back-end servers and response times are slow, the completeness condition might be, for example, receiving at least one response in a given time period. This service-oriented solution requires all three responses in order to construct the final message. Thus, here, the completeness condition is receiving all three responses.

## Determining Orchestration Boundaries

The solution must allow for requests from an IBM WebSphere MQ queue as well as through the Web service interface. Putting the service interface into one orchestration, the IBM WebSphere MQ input into another, and the bulk of the processing into a third keeps the external communication separated from the processing.

## Translating the Components into Orchestration Shapes

The patterns to translate into shapes will, in the final solution, be in a single orchestration. There are numerous ways to create a content-based router in BizTalk Server, including creating MessageBox subscriptions. In this case, the routing is simply based on whether or not the message contains the account number and either the customer name or zip code. An expression shape evaluates whether or not the message includes values for the required fields. A decision shape evaluates the variable set in the expression shape and selects the path through the orchestration.

The **CustomerService** orchestration uses the parallel shape to send and receive messages to the back-end applications so that it does not have to wait on one application finishing before making the next request. If the number of applications were to change frequently, or if the connections to the applications were unreliable, then the orchestration would need to translate the pattern differently.

In the solution the aggregator must combine elements from three response messages. Using the parallel shape ensures that the communication with the back-end systems is parallel. And, because the shape does not terminate until it receives all responses, the aggregator will always have all of the required messages. Thus, the orchestration uses a transform shape that maps elements from the three back-end response messages and the original request message to elements in the response message.

For a complete list of orchestration shapes, see [Orchestration Shapes](#) .

## Pattern Catalog for the Service Oriented Solution

The patterns in the service oriented solution include patterns of BizTalk Server-specific programming practices, as well as the enterprise integration patterns in preceding sections. The list in this section includes both kinds of patterns.

### Pattern Types

Entries described in the following topics briefly describe the pattern and point to other topics that describe how the solution uses the pattern. In the case of general patterns, such as a filter, entries point to more general topics.

### Aggregation Pattern

Aggregation is the pattern of receiving information from multiple sources and consolidating it into a single message. The service oriented solution combines credit information from three different sources into a single response. You can do aggregation several different ways, depending on the nature of the solution you're writing. In some cases, you may need to wait for all responses. In

other cases, such as in loan quotes, you may be willing to forego getting a response so long as you have some minimum number. The service oriented solution waits until it has all three responses in order to have a complete credit report to return. For more information, see *Translating the Patterns of the Service Oriented Solution* .

### **Calling Pipelines from Code Pattern**

You can now call pipelines from your code and orchestrations. This allows the re-use of pipelines and helps maintain the decoupling of an orchestration from the pipeline stages. For more information, see *Using Pipelines from Code* .

### **Caching Pattern**

Caching is a general strategy of storing information rather than retrieving it from a data store every time it is requested. The Enterprise Single Sign-On system proved to be a limiting factor in the solution. The solution caches the information and periodically refreshes the cache. For more information, see *Efficient Use of SSO* . The Business Process Management solution also caches the SSO information, though it uses a slightly different process. For more information, see *Using SSO Efficiently in the Business Process Management Solution* .

### **Content-Based Routing Pattern**

Content-based routing is determining the recipient of a message based on the some part of the content of the message. The service oriented solution uses a very simple form of content-based routing—a single decision shape in an orchestration sends the message one of two places. For more information, see "Translating the Components into Orchestration Shapes" in *Translating the Patterns of the Service Oriented Solution* . For another method of content-base routing, see *Content-Based Routing in BizTalk Explorer* .

### **Filter Pattern**

The filter pattern selects messages meeting particular criteria for processing. In BizTalk Server, the filter pattern almost always becomes a filter expression on a port. For more information about filters on ports, see *Using Filters to Receive Messages* .

### **Inline Invocation of Back-end Processes Pattern**

The inline version of the solution uses inline invocation of the back-end processes through custom assemblies. This has the benefit of greatly improved performance. It comes at the cost, however, of tightly coupling assemblies to the transport protocol. For more information, see *Inlining Back-end Invocation* .

### **Recipient List Pattern**

In an abstract sense, the service oriented solution implements a recipient list in that it sends messages to three different systems. In practical terms, the application hardwires the recipients in the Parallel Actions shape that constructs and sends the messages. For more information, see *Translating the Patterns of the Service Oriented Solution* .

## Service Interface Pattern

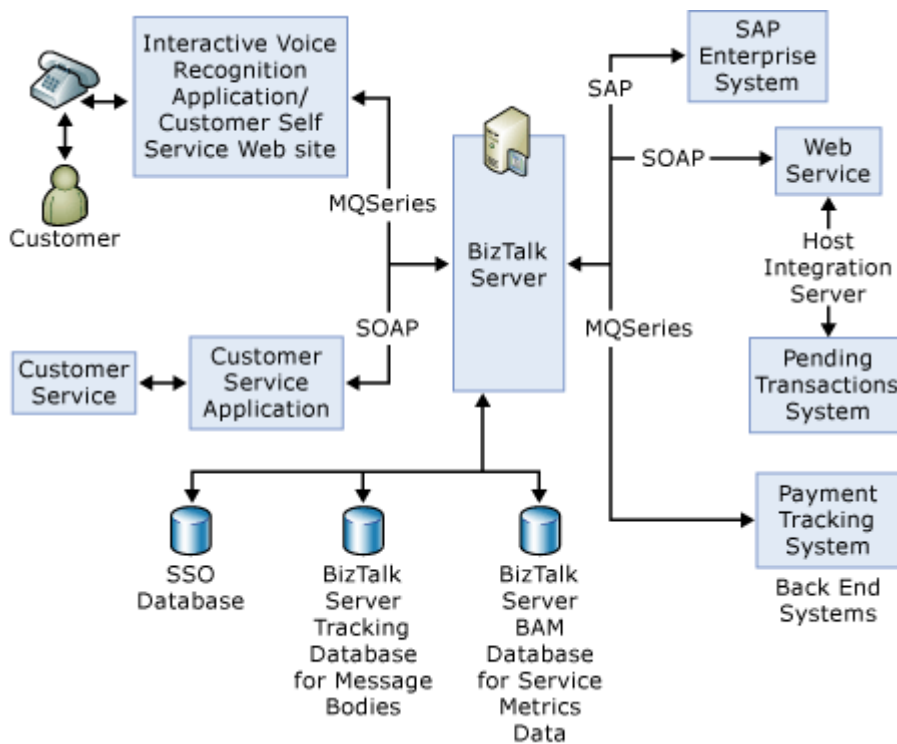
The service oriented solution presents itself as a Web service, only one of many ways a service may be done. For more information about using orchestrations as Web services, see Using Web Services .

## Translator Pattern

The enterprise pattern of a translator—that is, the conversion of a message from one form to another form—most often translates into a BizTalk Server map. For general information about BizTalk Server maps, see Creating Maps Using BizTalk Mapper .

## Components of the Service Oriented Solution

This section describes the major BizTalk Server components of the Service Oriented Solution. The following diagram shows the major components of the solution:



The Service Oriented solution has three versions of the orchestrations:

A version in which all three of the back-end applications are stubbed out

One in which all three back-end applications are invoked inline

A version that uses adapters to connect to the applications.

All versions of the orchestrations appear in the SDK\Senarios\BTSSoln\Orchestrations directory.



The inline version of the orchestrations provides the lowest latency time within the solution between requests and responses.

For information about the source files, see File Inventory for the Service Oriented Solution .

## **Orchestrations in the Service Oriented Solution**

Three orchestrations, CustomerServiceReceiveSend, CustomerServiceNativeRequestResponse, and CustomerService compose the bulk of the solution. The CustomerServiceReceiveSend and CustomerServiceNativeRequestResponse orchestrations act as front-ends that call the CustomerService orchestration. The CustomerService orchestration does most of the work—sending requests to the back-end applications, gathering the replies, combining the replies into a single message, and sending the message to the appropriate front-end orchestration. Because the front-end orchestrations call the CustomerService orchestration, the front-end orchestrations wait until the CustomerService orchestration finishes.

The solution exposes the CustomerServiceNativeRequestResponse orchestration as a Web service. The CustomerServiceReceiveSend orchestration takes messages from an MQSeries queue.

## **Back-end Applications**

The Service Oriented solution communicates with three back-end applications:

The PaymentTracker application returns a simulated list of recent payments. PaymentTracker reads the request from an MQSeries queue and sends the response to another MQSeries queue.

The PendingTransaction application reports the sum of transactions pending against the customer account. The application is a Web service that, in turn, uses Microsoft Host Integration Server (HIS) to communicate with a CICS/COBOL program on a mainframe computer.

The SAP application provides information about the customer's overall credit limit. The solution connects to the SAP application as a Web service. The application uses the SAP .NET Connector to communicate with a SAP system.

## **Pipelines**

The Service Oriented solution uses default pipelines except in two places: the receive pipeline for the CustomerServiceReceiveSend orchestration, and the CustomerService orchestration's send pipeline to the PaymentTracker. Both pipelines use custom components.

The receive pipeline for CustomerServiceReceiveSend includes a custom party resolution component, SSO Ticket Issuer Pipeline Component. The messages that the CustomerServiceReceiveSend orchestration receives do not have credentials. This simulates what would happen if the messages came from an Interactive Voice Response system. The custom pipeline component adds credentials using the service account of the BizTalk receive host.

In contrast, the messages the CustomerServiceNativeRequestResponse orchestration receives already have credentials. Because the virtual folder for the Web service is configured for integrated

security and the SOAP receive location is configured to integrate Enterprise Single Sign-On (SSO), the SOAP adapter generates a ticket for the message.

The other custom pipeline appears in the CustomerService send pipeline to the PaymentTracker application. The component, MQSeries Header Setter Pipeline Component, sets values for two MQSeries message header properties. The component sets the first, the Message Data Format (MQMD\_Format), to indicate the message is in the form of an MQCIH structure, a structure commonly used to communicate with CICS programs. The second, the format of the data itself within the MQCIH structure (MQCIH\_Format), is set to show the message is a string.

Using the MQCIH format enables you to pass the user ID and password in the MQCIH structure. SSO affiliate applications map the BizTalk application's Windows user ID to the Payment Tracking System's user IDs passed in the MQCIH structure.

## Client Application

The solution includes a client application written in C#. You can use the application to send requests as SOAP or MQSeries messages, and examine the results.

## Other Assemblies

The application includes several auxiliary assemblies not shown in the summary diagram above. The **Utilities** assembly utility functions for the solution.

The **ErrorHelper** assembly contains classes to translate error codes into messages, and to convert error messages to error codes.

The **ServiceLevelTracking** assembly includes helper methods using the Business Activity Monitoring (BAM) API to track service-level agreement data.

The **ConfigHelper** assembly contains helper methods to retrieve configuration values for the solution from the **SSOConfigStore** application.

## Implementation Highlights of the Service Oriented Solution

A solution solves a particular problem in a specific context. The Service Oriented solution is no exception and is specific to Microsoft BizTalk Server 2006 and the scenario. For more information about the Woodgrove Bank scenario, see Understanding the Service Oriented Solution .

While developing the scenario, several areas proved to be bottlenecks in reducing response times to an acceptable level. The first of these was the use of adapters. The adapters simply aren't fast enough to meet the required response times. For this reason, the inline version of the solution (the fastest version and the only one to meet all of the performance requirements) avoids adapters and calls the back-end systems directly. With three different back-end systems, that means potentially three different mechanisms to communicate with the back-end systems.

Another area that proved a performance problem was using Enterprise Single Sign-On (SSO) to store configuration information. To retain the convenience and universality of SSO while speeding up retrieval time, the solution uses a local cache for configuration values.

Another unusual element of the solution is calling pipelines directly from code. This allows reuse of the custom pipeline components while keeping the orchestration at least partially decoupled from the pipeline.

Finally, there are several BizTalk Server settings that you can change to squeeze out the last bit of speed from the solution.

### In This Section

- Inlining Back-end Invocation
- Efficient Use of SSO
- Using Pipelines from Code
- Tuning the Solution
- Decoupling Transport Type and Processing

## Inlining Back-end Invocation

The inline call version, of the full solutions, provides the fastest processing times. The inline version eliminates the overhead of persisting the request and response messages in the MessageBox database. In the adapter version, the message goes from the sending orchestration to the MessageBox. The host running the adapter picks up the message, and sends the message to the back-end process by again posting it to the message box.

The efficiency of inlining comes at a cost of binding the orchestration directly to the transport protocols of the back-end systems. In the inline version, rather than communicating through ports, the orchestration calls the back-end systems through three custom assemblies. If a back-end system transport changes, an assembly must be rewritten and recompiled. The following table describes the assemblies and their function:

Assembly Name	Back-end Connection
Microsoft.Samples.BizTalk.WoodgroveBank.PaymentTrackerCall	Uses MQSeries <b>get</b> and <b>put</b> message functions.
Microsoft.Samples.BizTalk.WoodgroveBank.PendingTransactionsCall	Invokes the Web service for the transaction system.
Microsoft.Samples.BizTalk.WoodgroveBank.SAPCall	Uses SAP remote function calls.

## Efficient Use of SSO

The service oriented solution uses Enterprise Single Sign-On (SSO) both to store configuration values and to handle credentials for the back-end systems. To reduce latency, the solution uses a local cache for the configuration values. The solution refreshes the cache every five minutes.

In many applications, the adapters handle the SSO operations—including getting an SSO ticket, redeeming the ticket, and using the credentials to gain access to the affiliate application. However, the inline version of the service oriented solution does not use adapters. It must use SSO from code.

This topic describes the caching mechanism used by the solution, as well as how the inline version of the solution uses SSO from code.

### Local Caching of Configuration Values

The service oriented solution uses two objects, **ConfigPropertyBag** and **ConfigParameters**, to handle the configuration values. The **ConfigPropertyBag** class holds the values and is used only by the **ConfigParameters** class. The **ConfigParameters** class is used by the other parts of the solution to retrieve the configuration parameters. Both classes are in the **Microsoft.Samples.BizTalk.WoodgroveBank.ConfigHelper** namespace.

The **ConfigPropertyBag** has the following methods:

Method	Description
Read	Retrieves a value for a given property.
Write	Assigns a value to a property.

The class uses an instance of a .NET **NameValueCollection** to hold the values. The two access methods implement the **IPropertyBag** interface from the **Microsoft.BizTalk.SSOClient.Interop** namespace. The **ConfigPropertyBag** class is an internal class and used only by the **ConfigParameters** class.

The application uses the **ConfigParameters** class to handle the SSO configuration values. The class has the following public methods and attributes:

Method or Attribute	Description
SSOConfigParameter	An enumeration for specifying configuration parameters.
GetConfigParameters	A method used to retrieve a value for a given parameter. Uses the SSOConfigParameter to indicate the parameter.

The **ConfigParameters** class uses the .NET Timer class and a delegate function to set up the refresh of the **ConfigPropertyBag**:

Notice that the constructor is static so that the static member variables can be initialized without having to create an instance of the class. The constructor creates instances of the SSO configuration store (**ISSOConfigStore**), the configuration property bag (**ConfigPropertyBag**), and a synchronization lock (**ReaderWriterLock**) used to control access to the **ConfigurationPropertyBag** during updates and reads. The constructor then uses **GetConfigInfo** to retrieve the SSO configuration values and to put them in the property bag. Finally, the constructor creates a **Timer** object that, after the specified interval, calls the delegate function, **cacheRefreshCallback**.

The **Timer** delegate function is relatively straightforward:

Notice that the delegate disables the timer so that it can run all the way through. Also notice the use of the locks to control access to the property bag. The **ReaderWriterLock** is the best choice here—it is designed for cases where there are more reads than writes. Notice also that the lock, **syncLock**, is static and declared at the class level so that it is visible to all threads.

## Using SSO from Code

When using single sign-on from code, the code must take on the role of the adapter: that is, retrieve the SSO ticket from the message, redeem the ticket to get the user name and password for the back-end system, and, finally, use the back-end system. The service-oriented solution does this through the **GetPendingTransactionsResponse** method of the **PendingTransactionsCaller** object.

The method appears as follows:

The method begins by retrieving configuration information, including the URL, for the back-end system and the name of the backend (affiliate) application.

To redeem the ticket, the method must extract the ticket and original requesting user name from the message. The message contains the ticket as one of the message context properties, **BTS.SSOTicket**. For more information about message context properties, see **Message Context Properties**. The method also extracts the **OriginatorSID** from the message context properties. With the ticket and the originator's name in hand, the method calls the **RedeemTicket** method on the ticket to retrieve the credentials.

The remainder of the code creates a .NET NetworkCredential cache for the credentials and calls the back-end Web service.

## Using Pipelines from Code

The inline version of the customer service orchestration (**CustomerService**) calls the payment tracking system directly. To prepare the sent message and process the received message, the orchestration calls the pipelines from code. This allows the reuse of the pipelines from the other scenarios versions. It also maintains the decoupling of the orchestration from the pipeline stages.

## Calling the Pipelines

To use the pipelines from code, you must create a message collection, add the message to process to the collection, create an empty message to receive the result message, and invoke the pipeline.

The following code from the **CustomerService** orchestration is in the **ConstructRequestMessageAfterSendPipeline** shape:

The **GetLastPaymentResponse** shape takes the message from the above code, sends it to the payment tracking system, and processes the returned message through the receive pipeline:

## Tuning the Solution

The orchestrations collect performance information about many parts of the application through the Business Activity Monitoring (BAM) API. You may find the data helpful in tuning the application. You can also, of course, use the built-in performance counters.

For more information about BAM, see **Business Activity Monitoring (BAM)** . For more information about how the solution uses BAM, see Monitoring the Service Oriented Solution with BAM . For more information about the BizTalk performance counters, see Orchestration Engine Performance Counters .

You can also tune the solution by adjusting values in the BizTalk Management database table `adm_ServiceClass`. For more information, see Performance tuning .

## Decoupling Transport Type and Processing

In a service oriented solution, a clear line often exists between the business processing and the specifics of transmitting and receiving messages. This enables you to change the business process or the messaging part of the solution independently.

The service oriented solution violates this design principle in one place. This section describes the situation, the possible alternatives, and the selected structure.

## Correlation and the MQSeries Adapter

In order to use the MQSeries adapter, you cannot use the standard BizTalk Server correlation identifiers. This is because the correlation identifier goes to an IBM back-end system that has its own system of correlation identifiers. Instead, you must use the **MQSeries.MQMD\_CorrelId** and **MQSeries.MQMD\_MsgID** properties. Using these properties potentially puts transport-specific information in the orchestration and, thus, in the business process.

One way to handle this dependency would be to use the BizTalk Server correlation identifier and use a custom pipeline component to translate the correlation identifier for MQSeries. This adds complexity to the scenario. In addition, if the transport changes, two pipeline components must be changed. And, ultimately, it relocates the dependency (in the pipeline component) rather than resolving it.

Another option would be to isolate the MQSeries-specific correlation handling to a separate orchestration and call that orchestration. This would preserve the independence of the business process. However, this introduces a compile-time dependency between the orchestrations. Modifying the transport requires recompiling both orchestrations (as, for example, in going from the stub to the adapter version of the solution). The call also adds to the response time for the solution.

Given the additional complexity and possible decrease in performance, it seemed simplest to use the MQSeries correlation directly in the orchestration.

For more information about the adapter and correlations in orchestrations, see [MQSeries Adapter - Correlating Messages with Orchestration](#).

## Monitoring the Service Oriented Solution with BAM

The solution monitors activity in all versions of the **CustomerService** orchestration using the Business Activity Monitoring (BAM) API. More specifically, it uses the new **OrchestrationEventStream** object.

### What is the OrchestrationEventStream Object?

The new **OrchestrationEventStream** object enables tracking in orchestrations and custom components without the inconsistencies and workarounds required by the other BAM objects. The **OrchestrationEventStream** is fully aware of transactions. All of its methods are static so that your orchestration does not need to create an instance of it.

For more information about the **OrchestrationEventStream** object, see XXX and T:Microsoft.BizTalk.Bam.EventObservation.OrchestrationEventStream.

Although the Tracking Profile Editor (TPE) is the preferred way of using BAM, TPE cannot capture the orchestration variable values, nor can it handle custom objects. The solution uses the BAM API to overcome these limitations.

For general information about BAM, see [Using Business Activity Monitoring](#). For information about the Tracking Profile Editor (TPE), see [Tracking Profile Editor](#). For best practices, see [Best Practices for Tracking Profiles](#).

### Wrapping the OrchestrationEventStream Object

The service oriented solution wraps the **OrchestrationEventStream** object with the **ServiceLevelTracking** object. The **ServiceLevelTracking** object provides application-specific milestone methods and hides some of the details of using **OrchestrationEventStream**.

As in **OrchestrationEventStream**, all the methods of **ServiceLevelTracking** are static. Thus, the orchestration or custom component does not need to create an instance of it. The method that begins tracking an activity, **TrackingBeginRequest**, returns a unique activity instance ID. All subsequent tracking events must be associated with this activity instance ID in order to capture the service level data correctly, because it is unique to the instance of the **CustomerService** orchestration.

## Versioning the Service Oriented Solution

The two orchestrations that act as front ends to the solution, **CustomerServiceReceiveSend** and **CustomerServiceNativeRequestResponse**, call the central, working orchestration, **CustomerService**. Orchestration calls depend on the version number of the assembly containing the orchestration. Because all three orchestrations are in the same assembly, there are no versioning issues.

In addition, the business process implemented by the orchestrations is a very short-lived request-response process that completes quickly. Thus, the question of versioning the business process does not arise in this solution—there are no different orchestrations in different assemblies with different versions.

However, the orchestrations do use the schemas in the schema assembly. If the schemas are revised and compiled into a different version, you must recompile the orchestrations with the newer version of the schema assembly.

## Scaling the Service Oriented Solution

The inline solution bypasses the MessageBox database when interacting with the back-end systems.

You can also add BizTalk Server processing servers to run orchestrations. This decreases the utilization of each server so that new requests can be processed quickly. You can also scale up the MessageBox server so that it has enough additional capacity to handle the message throughput.

However, the service oriented architecture solution does not benefit from having multiple MessageBox databases. Multiple MessageBoxes require distributed transaction coordinator (DTC) transactions. The transactions increase overall message latency.

You can decrease response time by eliminating the pipeline component that adds MQSeries header information. For more information, see [Implementation Highlights of the Service Oriented Solution](#).

## Deploying the Service Oriented Solution

Service Oriented Architecture (SOA) is an approach to building distributed systems. The service oriented solution demonstrates how several back-end systems using different protocols can be aggregated into a single service that clients can consume. This solution integrates services with an approach that guarantees delivery and performance characteristics.

There are three versions of the service oriented solution: adapter, inline, and stub.

The topics in this section describe how to install and run the service oriented solution on a single computer.



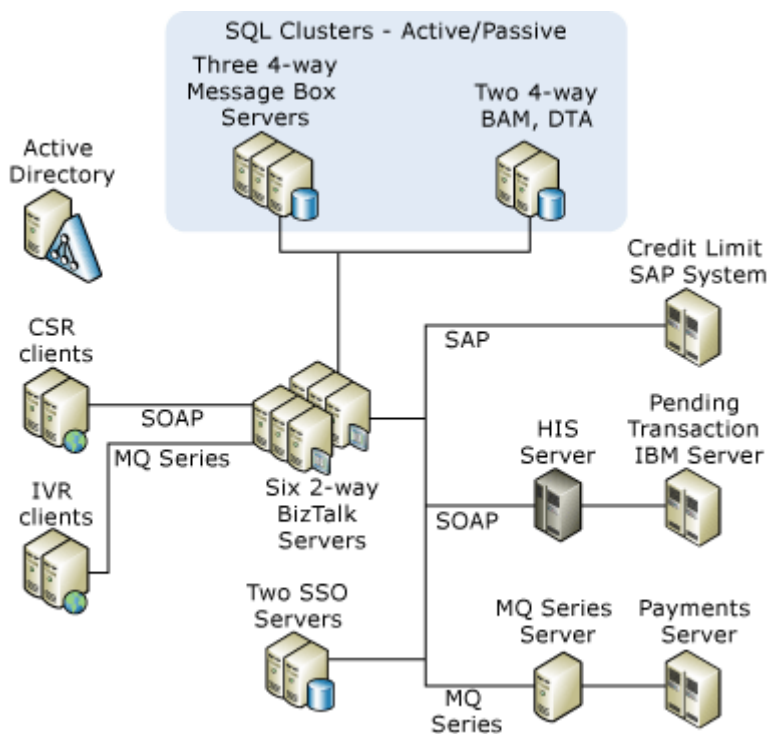
## Reader Guidance

This documentation assumes that you are familiar with Microsoft BizTalk Server and Microsoft Visual Studio 2005. It also assumes that you understand basic concepts about enterprise application integration and Web services. Additionally, it is recommended that you are familiar with how to build applications by using Visual Studio 2005 and that you are familiar with performing the following tasks: creating projects, setting references, and using the debug mode to debug and test your solution.

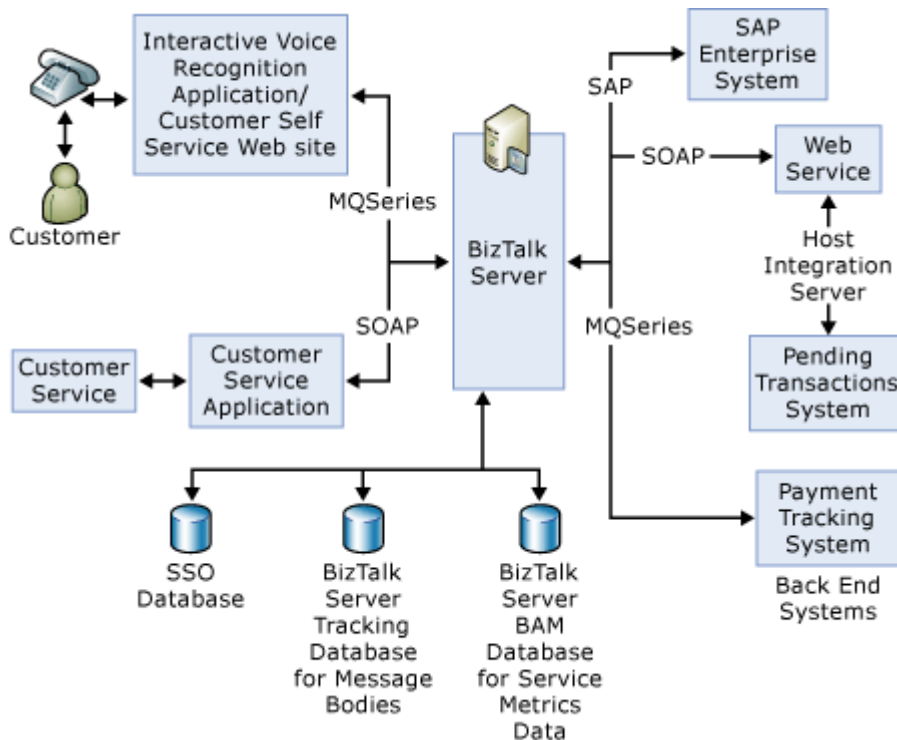
## Deployment Overview

In this scenario, you set up a credit card balance query process (as shown in the figure below). Customer Service Reps (CSRs) or Interactive Voice Response (IVR) clients provide credit card and credit card verification data to BizTalk Server. BizTalk Server, in turn, receives and assembles the response from the three back-end servers, which supply credit card payments, balance, and pending transactions. Connection credentials are stored in the Enterprise Single Sign-On (SSO) servers.

The following figure shows request-response with aggregation scenario base configuration.



The following diagram shows the major components of the solution.



### In This Section

- Developer Machine Setup for the SO Solution
- Deploying the SO Solution to Production Servers

### Before Installing the Service Oriented Solution

The following prerequisites must be available to install the stub version of the service-oriented solution on a single computer:

Microsoft Windows Server 2003

Microsoft BizTalk Server 2006

Microsoft SQL Server 2000 Service Pack 4

Microsoft Internet Information Services (IIS) 6.0 with ASP.NET enabled

Microsoft Visual Studio 2005

The latest version of IBM WebSphere MQ Client for Windows

For installing the inline and adapter version of the service oriented solution on a single computer:

Windows Server 2003

BizTalk Server 2006

SQL Server 2000 SP4

IIS 6.0 with ASP.NET enabled

MQSeries Server on the local computer or access to the computer running the MQSeries Server.  
For the inline version, MQSeries client APIs must be available on the BizTalk server running the solution's orchestrations.

IBM Mainframe with CICS and Transaction X if you are using a variation of the solution that requires a mainframe. In this case, the CICS application (COBOL code) must be installed on the mainframe and Microsoft Host Integration Server (HIS) is required to access the mainframe.

If you don't have a mainframe for the solution, you can modify the port binding to use the stub Web service for Pending Transactions. The Web service generates transactions locally to emulate the mainframe transactions. For more information about how to modify the port binding for the stub Pending Transactions Web service, see *How to Install the Inline and Adapter Versions of the Service Oriented Solution*.

Visual Studio 2005 with .NET Framework version 2.0 and, if using the mainframe, Visual Studio .NET 2003 with .NET Framework version 1.1.

Host Integration Server 2004 is required to connect to the mainframe.

Web server configured with a certificate for HTTPS connections.

## **How to Install the Stub Version of the Service Oriented Solution**

The following steps describe how to prepare your computer before you install the stub version of the service oriented solution, and then how to install the solution on your computer.

Prepare the computer for installing the stub version of the Service Oriented Solution

Install the IBM WebSphere MQ Client for Windows

Create the virtual directories in IIS for the Service Oriented Solution

Build the Service Oriented Solution

Create the Enterprise Single Sign-On (SSO) entries and values in the SSO database

Deploy the BAM definition for the Service Oriented Solution

Deploy the Service Oriented Solution

### **Prepare the computer for installing the stub version of the Service Oriented Solution**

## To prepare the computer for installing the stub version of the Service Oriented Solution

Make sure that the **Default Web Site** is configured to use ASP.NET 2.X.

Click **Start**, point to **All Programs**, point to **Administrative Tools**, and then click **Internet Information Services (IIS) Manager**.

In the **Internet Information Services (IIS) Manager**, expand **Web Sites**, right-click, the **Default Web Site**, select **Properties**, and then select **ASP.NET** tab.

Make sure that the **ASP.NET Version** is 2.X.

Click **Start**, point to **All Programs**, point to **Administrative Tools**, and then click **Services**. Using the **Services** console, make sure that the following services are running:

### World Wide Web Publishing

Click **Start**, point to **All Programs**, point to **Administrative Tools**, click **Computer Management** console, and then add the BizTalk service account to the local Administrators group.

If you installed Windows SharePoint Services, exclude the (root) of the **Default Web Site** from Windows SharePoint Services Managed Paths as follows: Click **Start**, point to **All Programs**, point to **Administrative Tools**, and then click **SharePoint Central Administration**.

Under **Virtual Server Configuration**, select **Configure virtual server settings**.

On the **Virtual Server List** page, click **Default Web Site**.

On the **Virtual Server Settings** page, click **Define managed paths**.

In the **Included Paths** section of the **Defined Managed Path** page, select **Root** and then click **Remove selected paths**.

At a command prompt, perform an IISReset.

Log off the computer, and then log on to the computer as the BizTalk service account.

Open a command prompt, type the following command, and then press ENTER to set the %BTSSolutionsPath% environment. Then, exit the command prompt.

```
setx    BTSSolutionsPath    "%ProgramFiles%\Microsoft    BizTalk    Server
2006\SDK\Scenarios"
```

## Install the IBM WebSphere MQ Client for Windows

### To install the IBM WebSphere MQ Client for Windows

Download the latest version of IBM WebSphere MQ Client for Windows.

Install IBM WebSphere MQ Client for Windows.

Add WebSphere MQ classes for the .NET assembly to the global assembly cache (GAC).

At the Visual Studio command prompt, navigate to the directory <IBM MQSeries Installation Directory>\bin.

Run the following command (make sure gacutil.exe is in the path environment):  
**gacutil.exe /i amqmdnet.dll**

## Create the virtual directories in IIS for the Service Oriented Solution

### To create the virtual directories in IIS for the Service Oriented Solution

In the **Internet Information Services (IIS) Manager**, right-click **Application Pools**, select **New**, and then select **Application Pool**.

On the **Add New Application Pool** dialog box, type **SSOStubAppPool** in the **Application pool ID** text box, and then click **OK**.

In the **Internet Information Services (IIS) Manager**, right-click the application pool that you just created, and then select **Properties**.

On the **Properties** page, click the **Identity** tab, select **Configurable**, enter the **User name** and **Password**, and then click **OK**. For this walkthrough use the same user account that the BizTalk service is using.

In the **Internet Information Services (IIS) Manager**, expand **Web Sites**, right-click the **Default Web Site**, point to **New**, and then click **Virtual Directory** to run **Virtual Directory Creation Wizard**.

Using the **Virtual Directory Creation Wizard**, create the following virtual directory for the proxy Web service for the adapter version:

Alias = Microsoft.Samples.BizTalk.WoodgroveBank.OrchProxy.Stub

PATH = <BizTalk Install Directory>\SDK\Scenarios\SO\BTSSoIn\OrchProxy\Stub

Access Permissions = Read, Run scripts

Using the **Virtual Directory Creation Wizard**, create the following virtual directory for the proxy Web service for the adapter version:

Alias = Microsoft.Samples.BizTalk.WoodgroveBank.StubSAP

PATH = <BizTalk Install Directory>\SDK\Scenarios\SO\BTSSoIn\StubWebServices\SAP

Access Permissions = Read, Run scripts

Using the **Virtual Directory Creation Wizard**, create the following virtual directory for the proxy Web service for the adapter version:

Alias = Microsoft.Samples.BizTalk.WoodgroveBank.StubPendingTransactions

PATH = <BizTalk Install  
Directory>\SDK\Scenarios\SO\BTSSoIn\StubWebServices\PendingTrans

Access Permissions = Read, Run scripts

Using the **Virtual Directory Creation Wizard**, create the following virtual directory for the proxy Web service for the adapter version:

Alias = Microsoft.Samples.BizTalk.WoodgroveBank.StubPaymentTracker

PATH = <BizTalk Install  
Directory>\SDK\Scenarios\SO\BTSSoIn\StubWebServices\PaymentTrack

Access Permissions = Read, Run scripts

In the **Internet Information Services (IIS) Manager**, expand **Web Sites**, expand the **Default Web Site**, right-click Microsoft.Samples.BizTalk.WoodgroveBank.OrchProxy.Stub, click **Properties**, and then modify the settings as follows:

On the **Virtual directory** tab, set the **Application Pool** to **SSOStubAppPool** you just created.

Click **Directory Security** tab, click **Edit** in the **Authentication and access control** group box, select **Only Integrated Windows Authentication enabled**, and then clear other **Authentication access** checkboxes. Click **OK** to exit.

In the **Internet Information Services (IIS) Manager**, expand **Web Sites**, expand the **Default Web Site**, right-click Microsoft.Samples.BizTalk.WoodgroveBank.StubSAP, click **Properties**, and then modify the settings as follows:

On the **Virtual directory** tab, set the **Application Pool** to **SSOStubAppPool** you just created.

Click **Directory Security** tab, click **Edit** in the **Authentication and access control** group box, and then select **Enable Anonymous Access**. Click **OK** to exit.

In the **Internet Information Services (IIS) Manager**, expand **Web Sites**, expand the **Default Web Site**, right-click Microsoft.Samples.BizTalk.WoodgroveBank.StubPendingTransactions, click **Properties**, and then modify the settings as follows:

On the **Virtual directory** tab, set the **Application Pool** to **SSOStubAppPool** you just created.

Click **Directory Security** tab, click **Edit** in the **Authentication and access control** group box, and then select **Enable Anonymous Access**. Click **OK** to exit.

In the **Internet Information Services (IIS) Manager**, expand **Web Sites**, expand the **Default Web Site**, right-click **Microsoft.Samples.BizTalk.WoodgroveBank.StubPaymentTracker**, click **Properties**, and then modify the settings as follows:

On the **Virtual directory** tab, set the **Application Pool** to **SSOStubAppPool** you just created.

Click **Directory Security** tab, click **Edit** in the **Authentication and access control** group box, and then select **Enable Anonymous Access**. Click **OK** to exit.

## **Build the Service Oriented Solution**

### **To build the Service Oriented Solution**

Click **Start**, point to **All Programs**, point to **Microsoft Visual Studio 2005**, point to **Visual Studio Tools**, and then click **Visual Studio 2005 Command Prompt**.

At the **Visual Studio 2005 Command Prompt**, change the directory to the **%BTSSolutionsPath%\SO\BTSSoln** folder, and then run the following command to build the stub version of service-oriented solution.

**SetupBTSSoln.bat**

## **Create the Enterprise Single Sign-On (SSO) entries and values in the SSO database**

### **To create the Enterprise Single Sign-On (SSO) entries and values in the SSO database**

Open a command prompt, change the current directory to the **%BTSSolutionsPath%\SO\BTSSoln\Scripts**, and then run the following command to set the PATH environment for the Enterprise Single Sign-On folder.

**Set PATH=%PATH%;%ProgramFiles%\Common Files\Enterprise Single Sign-On"**

At the command prompt, change the directory to the **%BTSSolutionsPath%\SO\BTSSoln\Scripts** folder, open **ConfigStoreApp.xml** using Notepad, and then review the contents of the file.

At the command prompt, run the following command to create the SSO configuration store application.

**ssomanage -createapps ConfigStoreApp.xml**

At the command prompt, open **SetConfigValuesInSSO.cmd** using Notepad, and then review the contents of the file

At the command prompt, type **SetConfigValuesInSSO.cmd**, and then press ENTER to store the values in the SSO configuration store application.

At the command prompt, run the following command to enable tickets in SSO:

**ssomanage -tickets yes yes**

## **Deploy the BAM definition for the Service Oriented Solution**

### **To deploy the BAM definition for the Service Oriented Solution**

At a command prompt, type the following command, and then press ENTER. This sets the path to find the BAM utility:

**SET PATH=%PATH%;%ProgramFiles%\ "Microsoft BizTalk Server 2006\Tracking"**

At the command prompt, change the directory to the %BTSSolutions%\SO\BTSSoln\BAM folder, and type the following command, and then press ENTER:

**bm deploy-all -DefinitionFile:ServiceLevelTracking.xml**

## **Deploy the Service Oriented Solution**

### **To deploy the Service Oriented Solution**

Open a command prompt, change the directory to the %BTSSolutionsPath%\SO\BTSSoln\Scripts folder, type the following command, and then press ENTER:

**DeployStubBinding.cmd**

At the command prompt, run the following command to start the orchestrations for the stub version

**Startstub.vbs**

## **How to Install the Inline and Adapter Versions of the Service Oriented Solution**

The following steps describe how to prepare the computer for installing the inline and adapter versions of the service oriented solution, and how to install the solution on this computer.

Prepare the computer for installing the adapter and inline versions of the Service Oriented Solution

Remove the stub version of the Service Oriented Solution

Prepare the back-end systems for the Service Oriented Solution to access

Configure the Web server for Secure Socket Layers (SSL)

Create the Web services for the back-end systems

Create the TI component for the Service Oriented Solution



Create the virtual directories for the orchestration Web services

Build the Service Oriented Solution

Create the SSO affiliate applications

Deploy the BAM definition file for the Service Oriented Solution

Deploy the Service Oriented Solution

Configure the stub Pending Transactions Web Services when a mainframe is not available

Start the Service Oriented Solution

### **Prepare the computer for installing the adapter and inline versions of the Service Oriented Solution**

#### **To prepare the computer for installing the adapter and inline versions of the Service Oriented Solution**

if you installed Windows SharePoint Services, exclude the (root) of the Default Web Site from Windows SharePoint Services Managed Paths as follows: Click **Start**, point to **All Programs**, point to **Administrative Tools**, and then click **SharePoint Central Administration**.

Under **Virtual Server Configuration**, select **Configure virtual server settings**.

On the **Virtual Server List** page, click **Default Web Site**.

On the **Virtual Server Settings** page, click **Define managed paths**.

In the **Included Paths** section of the **Defined Managed Path** page, select **Root** and then click **Remove selected paths**.

At a command prompt, perform an IISReset.

Click **Start**, point to **All Programs**, point to **Administrative Tools**, click **Computer Management** console, and then add the BizTalk service account to the local Administrators group.

Log off the computer, and then log on to the computer as the BizTalk service account.

At a command prompt, type the following command, and then press ENTER to set the %BTSSolutionsPath% environment variable. Then, exit the command prompt.

```
setx      BTSSolutionsPath      "%ProgramFiles%\Microsoft      BizTalk      Server
2006\SDK\Scenarios"
```

### **Remove the stub version of the Service Oriented Solution**

## To remove the stub version of the Service Oriented Solution

Open the **BizTalk Server Administration** console as follows: Click **Start**, point to **All Programs**, point to **BizTalk Server 2006**, and then click **BizTalk Server Administration**.

In the **BizTalk Server Administration** console, expand **BizTalk Server 2006 Administration**, expand **BizTalk Group**, expand **Applications**, right-click **BTSScn.SO.CustomerService**, and then click **Stop**. In the **Stop Application** dialog box, select **Full Stop - Terminate Instances**, and then click **Stop**.

Open a command prompt, type the following command, and then press ENTER. This command changes the default script host to CScript.exe:

```
cscript /H:CScript
```

At the command prompt, change the current directory to %BTSSolutionsPath%\SO\BTSSoln\Scripts folder, type the following command, and then press ENTER:

```
UnEnlistStub.vbs
```

At the command prompt, type the following command, and then press ENTER:

```
UndeployStub.vbs
```

At the command prompt, run the following command

```
SET PATH=%PATH%;%ProgramFiles%\ "Microsoft BizTalk Server 2006\Tracking"
```

This sets the path to find the BAM utilities.

At the command prompt, change the directory to %BTSSolutionsPath%\SO\BTSSoln\BAM, and then run the following command:

```
bm remove-all -DefinitionFile:ServiceLevelTracking.xml
```

At the command prompt, change the directory to <Enterprise Single Sign-On Install Directory>, and then run the following command:

```
ssomanage -tickets no no
```

At the command prompt, run the following command to delete the WoodgroveBank.CustomerService SSO Affiliated application:

```
ssomanage -deleteapp WoodgroveBank.CustomerService
```

At the command prompt, run the following commands to delete the Web sites used by the stub version.

```
iisvdir /delete W3SVC/1/ROOT/Microsoft.Samples.BizTalk.WoodgroveBank.OrchProxy.Stub
```

```
iisvdir /delete W3SVC/1/ROOT/Microsoft.Samples.BizTalk.WoodgroveBank.StubSAP
```

```
iisvdir /delete  
W3SVC/1/ROOT/Microsoft.Samples.BizTalk.WoodgroveBank.StubPendingTransactions
```

```
iisvdir /delete  
W3SVC/1/ROOT/Microsoft.Samples.BizTalk.WoodgroveBank.StubPaymentTracker
```

Start Internet Information Services (IIS) Manager as follows: Click **Start**, point to **All Programs**, point to **Administration Tools**, and then click **Internet Information Services (IIS) Manager**.

Expand the **Application Pools**, right-click the application pool you created for the previous Web applications, click **Delete**, and then click **OK** in the confirmation dialog box.

Click **Start**, point to **Control Panel**, click **Add or Remove Programs**, and then uninstall the IBM WebSphere MQ Client for Windows.

Click **Start**, point to **All Programs**, point to **Microsoft Visual Studio 2005**, point to **Visual Studio Tools**, and then click **Visual Studio 2005 Command Prompt**.

At the **Visual Studio 2005 Command Prompt**, run the following command to delete the amqmdnet.dll you installed for the stub version.

```
gacutil /u amqmdnet
```

### Prepare the back-end systems for the Service Oriented Solution to access

#### To install the required applications for the back-end systems for the Service Oriented Solution to access

Install IBM WebSphere MQ for Windows Version 5.3 Server on the local computer.

Keep all the default settings. Set up the **Default Configuration** at the end of the **Prepare WebSphere MQ Wizard**. The queue manager is named as QM\_<your computer name>.

Install the Fix Pack 10 (CSD10). Keep all the default settings.

Install the MQSeries Agent.

Rerun the Microsoft BizTalk Server 2006 setup program.

On the **Program Maintenance** page, select **Modify**, and then click **Next**.

On the **Component Installation** page, expand the **Additional Software** node, and then select **MQSeries Agent**.

On the **Completion** page, make sure that **Launch BizTalk MQSeries Agent Configuration Wizard** is not selected.

Open a **Visual Studio 2005 Command Prompt**, change the directory to the <IBM MQSeries Installation Directory>\bin folder, and then run the following command:

```
gacutil /i amqmdnet.dll
```

Install Microsoft Visual Studio .NET 2003 if you want to install Microsoft Host Integration Server 2004 to access the mainframe. In the setup program, on the **Options** page, select **Visual C# .NET**, and then clear other components checkboxes. You don't need to install other components than the **Visual C# .NET**.

Install and configure Microsoft Host Integration Server 2004 if you have a mainframe to be accessed. Keep all the default settings.

### **To create the MQSeries queues**

Open the WebSphere MQ Explorer, expand **Queue Managers**, and then expand the queue manager in which you want to create the queues. Typically, a queue manager is named as QM\_<your computer name>.

In the WebSphere MQ Explorer, right-click **Queues**, point to **New**, click **Local Queue**, and then create the following local queues for the adapter version of the solution:

AdapterSOAInputQueue

AdapterSOAOutputQueue

Repeat the previous step to create the following local queues for the inline version:

InlineSOAOutputQueue

InlineSOAInputQueue

Repeat the previous step to create the following local queues for the Payment Tracker simulator. (The Payment Tracker simulator is used in both the adapter and inline versions):

LastPaymentsInputQueue

LastPaymentsOutputQueue

### **To complete configuration of the MQSeries adapter**

Click **Start**, point to **Programs**, point to **Microsoft BizTalk Server 2006**, and then click **BizTalk MQSeries Agent Configuration Wizard**.

On the **Welcome** page, click **Next**.

On the **Application Identity** page, select **This User**, and then enter the user name and password. The COM+ application for the MQSeries Agent will run under this user account. For this walkthrough, use the same user account that the BizTalk service is using. If it is not, the user accounts for BizTalk services hosting the MQSeries Adapter must be added to the **CreatorOwner** role of the COM+ application.

Click **Yes** on the **MQSConfigWiz** dialog box, if prompted that the user account that you entered in the previous step has the administrative privilege.

On the **Name of Role** page, click **Next**.

On the **Creating the MQSAgent COM+ Application** page, click **Next**, and then click **Finish** on the **Completion** page.

### To configure the mainframe CICS application

Using Notepad, open the bizcbl.txt and its "copy book" (MainFrameProgramVTCS2Description.txt) in the %BTSSolutionsPath%\SO\MFAccess\HISTIComponent folder, and then review the contents.

Bizcbl.txt includes the COBOL procedure returning the randomized account statements from account number input.

MainFrameProgramVTCS2Description.txt contains COMMAREA which describes the input and output data information. COMMAREA is a block of contiguous memory used to pass data back and forth between called and calling programs.

Copy the COBOL code to the host by method like FTP.

Compile the COBOL code and copy book. The following code shows a sample of Job Control Language (JCL) for the COBOL compiler.

Link edit the compiled source to create the executable. The following code shows a sample of JCL for COBOL link edit.

Configure the CICS mainframe application.

In this step, the mainframe systems programmer or CICS developer must install TCP/IPSERVICE, Session, Connection, Transaction, and Program resource definitions.

You should consult with mainframe administrators to get an IP address, port number, and a Link to Program name that you can access.

### Configure the Web server for Secure Socket Layers (SSL)

#### To install Certificate Services

Click **Start**, point to **Control Panel**, and then click **Add or Remove Programs**.

In the **Add or Remove Programs** dialog box, click **Add/Remove Windows Components**.

In the **Windows Components Wizard**, select the **Certificate Services**, click **Next**, and then follow the on-screen instructions to complete the installation.

### **To create a certificate request**

In the **Internet Information Services (IIS) Manager**, expand **Web Sites**, right-click the **Default Web Site**, click **Properties**, click the **Directory Security** tab, and then click **Server Certificate**.

On the **Welcome** page of the **Web Server Certificate Wizard**, click **Next**.

On the **Service Certificate** page, select **Create a new certificate**, and then click **Next**.

On the **Delayed or Immediate Request** page, click **Prepare the request now, but send it later**, and then click **Next**.

On the **Name and Security Settings** page, keep all the default settings, and then click **Next**.

On the **Organization Information** page, type your company's organization and organizational unit names, and then click **Next**.

On the **Your Site's Common Name** page, type your computer name in the **Common name** box, and then click **Next**.

On the **Geographical Information** page, fill out your geographical information, and then click **Next**.

On the **Certificate Request File Name** page, type **c:\certreq.txt** in the **File name** box, and then click **Next**.

On the **Request File Summary** page, click **Next**, and then click **Finish** on the **Completion** page.

### **To submit the certificate request to the Certification Authority**

In Internet Explorer, visit the following Web site:

**<http://localhost/certsrv>**

On the **Welcome** page, click **Request a Certificate**, and then click **Advanced certificate request** on the **Request a Certificate** page.

## To issue a certificate using Certification Authority management tool

Click **Start**, point to **Administrative Tools**, and then click **Certification Authority**.

In the **Certification Authority** console, expand your certification authority's name, expand the **Pending Request**, right-click the certificate request that you submitted in the previous step, point to **All Tasks**, and then click **Issue**.

Close the **Certification Authority** console.

## To download the certificate to the Web server

In Internet Explorer, visit the following Web site:

**<http://localhost/certsrv>**

On the **Welcome** page, click **View the status of a pending certificate request**.

## To install the certificate to the Web server

In the **Internet Information Services (IIS) Manager**, expand **Web Sites**, right-click the **Default Web Site** for which you created the certificate request, and then click **Properties**.

On the **Properties** dialog box, click the **Directory Security** tab, and then click **Server Certificate**.

On the **Welcome** page of the **Web Server Certificate Wizard**, click **Next**.

On the **Pending Certificate Request** page, select **Process the pending request and install the certificate**, and then click **Next**.

On the **Process a Pending Request** page, type **c:\certnew.cer** in the **Path and file name** text box, and then click **Next**.

Click **Next** on the **SSL Port** page, click **Next** on the **Certificate Summary** page, and then click **Finish** on the **Confirmation** page.

## Create the Web services for the back-end systems

### To create a new IIS application pool for the Pending Transaction Web services

In the **Internet Information Services (IIS) Manager**, right-click **Application Pools**, select **New**, and then select **Application Pool**.

On the **Add New Application Pool** dialog box, enter the **Application pool ID** (any value), and then click **OK**.

In the **Internet Information Services (IIS) Manager**, right-click the application pool that you just created, and then select **Properties**.

On the **Properties** page, click the **Identity** tab, select **Configurable**, enter the **User name** and **Password**, and then click **OK**. For this walkthrough, use the same user account that the BizTalk service is using.

### To create the PendingTransactions Web service for runtime

In the **Internet Information Services (IIS) Manager**, expand **Web Sites**, right-click the **Default Web Site**, point to **New**, and then click **Virtual Directory** to run **Virtual Directory Creation Wizard**.

Using the **Virtual Directory Creation Wizard**, create the following virtual directory for the stub SAP Web service:

Alias = Microsoft.Samples.BizTalk.WoodgroveBank.PendingTransactions

PATH = <BizTalk Install Directory>\SDK\Scenarios\SO\MFAccess\PendingTransactions

Access Permissions = Read, Run scripts

In the **Internet Information Services (IIS) Manager**, expand **Web Sites**, expand the **Default Web Site**, right-click Microsoft.Samples.BizTalk.WoodgroveBank.PendingTransactions, and then click **Properties**.

In the **Directory Security** tab, click **Edit** to modify **Authentication and access control**. Select **Basic authentication (password is sent in clear text)**, and clear other **Authentication access** checkboxes. Click **OK** to close the **Authentication Methods** dialog box.

In the **Directory Security** tab, click **Edit** under the **Secure Communication** group box, and then check **Require secure channel (SSL)** in the **Secure Communications** dialog box.

In the **Virtual Directory** tab, set the **Application Pool** to the application pool that you created in the procedure "To create a new IIS application pool for the Pending Transaction Web services".

### To create the PendingTransactions Web service for development environment

In the **Internet Information Services (IIS) Manager**, expand **Web Sites**, right-click the **Default Web Site**, point to **New**, and then click **Virtual Directory** to run **Virtual Directory Creation Wizard**.

Using the **Virtual Directory Creation Wizard**, create the following virtual directory for the stub SAP Web service:

Alias = PendingTransactions

PATH = <BizTalk Install Directory>\SDK\Scenarios\SO\MFAccess\PendingTransactions



Access Permissions = Read, Run scripts

In the **Internet Information Services (IIS) Manager**, expand **Web Sites**, expand the **Default Web Site**, right-click PendingTransactions, and then click **Properties**.

In the **Directory Security** tab, click **Edit** to modify **Authentication and access control**. Select **Enable anonymous access**. Click **OK** to exit.

In the **Virtual Directory** tab, set the **Application Pool** to the application pool that you created in the procedure "To create a new IIS application pool for the Pending Transaction Web services".

### To create the Stub SAP Web service

In the **Internet Information Services (IIS) Manager**, expand **Web Sites**, right-click the **Default Web Site**, point to **New**, and then click **Virtual Directory** to run **Virtual Directory Creation Wizard**.

Using the **Virtual Directory Creation Wizard**, create the following virtual directory for the stub SAP Web service:

Alias = Microsoft.Samples.BizTalk.WoodgroveBank.StubSAP

PATH = <BizTalk Install Directory>\SDK\Scenarios\SO\BTSSoln\StubWebServices\SAP

Access Permissions = Read, Run scripts

In the **Internet Information Services (IIS) Manager**, expand **Web Sites**, expand the **Default Web Site**, right-click Microsoft.Samples.BizTalk.WoodgroveBank.StubSAP, click **Properties**, and then modify the settings as follows:

In the **Virtual directory** tab, set the **Application Pool** to <YourAppPool> that you created in the procedure "To create a new IIS application pool for the Pending Transaction Web services".

In the **Directory Security** tab, click **Edit** in the **Authentication and access control** group box, and then select **Enable anonymous access**. Click **OK** to exit.

### Create the TI component for the Service Oriented Solution

#### To create a COM+ application for the TI component

At a command prompt, run %systemroot%\system32\com\comexp.msc.

In **Component Services** console, expand **Component Services**, expand **Computers**, expand **My Computer**, right-click **COM+ Application**, point to **New**, and then click **Application**.

On the **Welcome** page, click **Next**, and then click **Create an empty application** on the **Install or Create a New Application** page.

Type **BTSScn SO TI Component** in the **Enter a name for the new application** box, select **Server application** as **Activation type**, and then click **Next**.

In the **Account** group box of the **Set Application Identity** page, select **This user**, and then type the user name and password in the **User** and **Password** boxes. The new COM+ application will run under this user account. This user account must be a member of local HIS Runtime Users group. For this walkthrough, use the same user account that the BizTalk service is using.

On the **Add Application Roles** page, click **Next**.

On the **Add Users to Roles** page, expand **CreatorOwner**, click **Users**, and then click **Add**.

On the **Select Users or Groups** dialog box, select a user account that will be used for accessing the mainframe. For this walkthrough, add UserID local account.

On the **Completion** page, click **Finish**.

### To create a Remote Environment to access the mainframe

Click **Start**, point to **All Programs**, point to **Microsoft Host Integration Server 2004**, and then click **TI Manager**.

In the **TI Manager** console, click **Transaction Integrator (Configuration)**, expand **Windows Initiated Processing**, right-click **Remote Environments**, point to **New**, and then click **Remote Environment**.

On the **Welcome** page, click **Next**.

On the **Configure a New Remote Environment** page, type the **Remote Application Name**, and then click **Next**. For this walkthrough, use Mainframe\_TCP for the name.

On the **Configure Host Environment and Programming Model** page, select **CICS** for the **Target host** and **ELM Link** for the **Programming model**, and then click **Next**.

On the **Configure Endpoint TCP/IP** page, type the IP address for your mainframe in the **IP/DNS address** box, and then click **Edit** to add the port number. Your HIS COM will access the transactions through the endpoint address.

On the **Completion** page, click **Finish**.

### To create the TI Component for the Service Oriented Solution

Click **Start**, point to **All Programs**, point to **Microsoft Host Integration Server 2004**, and then click **TI Manager**.

In the **TI Manager** console, click **Transaction Integrator (Configuration)**, click **Windows Initiated Processing**, and then click **Objects**. Right-click **Objects**, click **New**, and then click **Object**.

On the **Welcome** page, click **Next**.

On the **Specify Or Locate An Object** page, click **Browse**, choose the SOHISTIUsingCOM.TLB in the %BTSSolutionsPath%\SO\MFAccess\HISTIComponent folder, and then click **Next**.

On the **Define Environment Characteristics for The COM Object** page, select **BTSScn SO TI Component** for the **COM+ application**, and then click **Next**.

On the **Define Remote Environment** page, select the remote environment that you created in the previous procedure for the **Remote environment**, and then click **Next**.

On the **Creation of WIP Objects** page, click **Next**, and then click **Finish** on the **Completion** page.

### To test the connectivity to the mainframe

In Windows Explorer, browse to the %BTSSolutionsPath%\SO\MFAccess\HISTISimpleTester folder, and then double-click the Interop.SOHISTIUsingCOM.dll.reg file. This adds registry values for the HISTISimpleTester application to call the TI component through the Runtime Callable Wrapper (RCW).

In Windows Explorer, browse to the %BTSSolutionsPath%\SO\MFAccess\ folder, and then run SetupMFAccess.bat.

In Windows Explorer, navigate to the %BTSSolutionsPath%\SO\MFAccess\HISTISimpleTester\bin\Debug folder, and then run BTSScnSOHISTIComponentSimpleTester.exe.

In the HISTISimpleTester application, click **Call Mainframe Program - Using COM**. It returns five records from the COBOL application running on the mainframe.

### Create the virtual directories for the orchestration Web services

#### To create the virtual directories for the orchestration Web services

In the **Internet Information Services (IIS) Manager**, right-click **Application Pools**, select **New**, and then select **Application Pool**.

On the **Add New Application Pool** dialog box, enter the **Application pool ID** (any value), and then click **OK**.

Right-click the application pool that you just created, and then select **Properties**.

On the **Properties** page, click the **Identity** tab, select **Configurable**, enter the **User name** and **Password**, and then click **OK**. For this walkthrough use the same user account that the BizTalk service is using.

In the **Internet Information Services (IIS) Manager**, expand **Web Sites**, right-click the **Default Web Site**, point to **New**, and then click **Virtual Directory** to run **Virtual Directory Creation Wizard**.

Using the **Virtual Directory Creation Wizard**, create the following virtual directory for the proxy Web service for the adapter version:

Alias = Microsoft.Samples.BizTalk.WoodgroveBank.OrchProxy.Adapter

PATH = <BizTalk Install Directory>\SDK\Scenarios\SO\BTSSoln\OrchProxy\Adapter

Access Permissions = Read, Run scripts

In the **Internet Information Services (IIS) Manager**, expand **Web Sites**, expand the **Default Web Site**, right-click Microsoft.Samples.BizTalk.WoodgroveBank.OrchProxy.Adapter, click **Properties**, and then modify the settings as follows:

In the **Virtual directory** tab, set the **Application Pool** to <YourAppPool> that you created in the previous step.

In the **Directory Security** tab, click **Edit** in the **Authentication and access control** group box, select **Only Integrated Windows Authentication enabled**, and then clear other **Authentication access** checkboxes. Click **OK** to exit.

In the **Internet Information Services (IIS) Manager**, expand **Web Sites**, right-click the **Default Web Site**, point to **New**, and then click **Virtual Directory** to run **Virtual Directory Creation Wizard**.

Using the **Virtual Directory Creation Wizard**, create the following virtual directory for the proxy Web service for the inline version:

Alias = Microsoft.Samples.BizTalk.WoodgroveBank.OrchProxy.Inline

PATH = <BizTalk Install Directory>\SDK\Scenarios\SO\BTSSoln\OrchProxy\Inline

Access Permissions = Read, Run scripts

In the **Internet Information Services (IIS) Manager**, expand **Web Sites**, expand the **Default Web Site**, right-click Microsoft.Samples.BizTalk.WoodgroveBank.OrchProxy.Inline, click **Properties**, and then modify the settings as follows:

On the **Virtual directory** tab, set the **Application Pool** to <YourAppPool> you just created.

Click **Directory Security** tab, click **Edit** in the **Authentication and access control** group box, select **Only Integrated Windows Authentication enabled**, and then clear other **Authentication access** checkboxes. Click **OK** to exit.

## Build the Service Oriented Solution

### To build the Service Oriented solution

At a command prompt, change the directory to the %BTSSolutionsPath%\SO\BTSSoln, type **SetupBTSSoln.bat**, and then press ENTER. The SetupBTSSoln.bat performs the following tasks:

- Creates a unique strong name key (SNK) for signing the assemblies of the SO Solution

- Extracts the public key token from the SNK and updates the binding files with the public token.

- Builds the SO solution.

- Builds the SSOApplicationConfig in the %BTSSolutionsPath%\Common folder.

## Create the SSO affiliate applications

### To create the SSO affiliate applications

Open a command prompt, and then change the directory to the %BTSSolutionsPath%\SO\BTSSoln\Scripts folder.

At the command prompt, open the PendTransAffApp.xml using Notepad, and review it. No changes to this file are necessary.

At the command prompt, open the PendTransUserMap.xml file using Notepad, and then make the following edits:

At the command prompt, open the PmntTrckAffApp.xml file using Notepad, and review the contents of the file. No changes to this file are necessary.

At the command prompt, open the PmntTrckUserMap.xml file using Notepad, and then make the following edits:

At the command prompt, open the ConfigStoreApp.xml file using Notepad, and then review the contents of the file. This file defines the configuration store application in SSO that the scenario uses to keep configuration parameters. Some of the configuration parameters include the Timeout value when communicating with SAP (for both the adapter and inline versions) and the name of the queue manager and queues to use when using the inline version. No changes to this file are necessary.

At the command prompt, open the SetConfigValuesInSSO.cmd file using Notepad, review and change the contents of the file as the following table.

Parameter	Default Value
SAPAdapterTimeout	20000
SAPInlineTimeout	20000
SAPInlineHostName	<User Specified>
SAPInlineClientNumber	<User Specified>
SAPInlineSystemNumber	<User Specified>
SAPInlineUserName	<User Specified>
SAPInlinePassword	<User Specified>
PendingTransactionsAdapterTimeout	20000
PendingTransactionsInlineTimeout	20000
PendingTransactionsInlineURL	https://<your name>/Microsoft.Samples.BizTalk.WoodgroveBank.PendingTransactions

PendingTransactionsInlineSSOAffiliateApp	WoodgroveBank.PendingTransactions
PaymentTrackingAdapterTimeout	20000
PaymentTrackingInlineTimeout	20000
PaymentTrackingInlineQManger	<User Specified> (Typically QM_<your computer name>).
PaymentTrackingInlineMQChannelDefinition	" " (need to enter the two double quotes).
PaymentTrackingInlineRequestQueue	LastPaymentsInputQueue

PaymentTrackingInlineResponseQueue	LastPaymentsOutputQueue
PaymentTrackingInlineSSOAffiliateApp	WoodgroveBank.PaymentTracker
StubSAPWebServiceURL	http://localhost/Microsoft.Samples.BizTalk.WoodgroveBank.StubSAP

At the command prompt, run the following command to set the PATH environment:

```
SET PATH=%PATH%;"%CommonProgramFiles%\Enterprise Single Sign-On"
```

At the command prompt, run the CreateInitialConfigInSSO.cmd. It creates the SSO Affiliate Applications, the SSO configuration store application, and the user mappings for the affiliate applications. Then, it executes the SetConfigValuesInSSO.cmd to store configuration values in the SSO configuration store application.

At the command prompt, run the following command to set the user credential for the Pending Transactions affiliate application. Use the **<DomainName>** and **<UserID>** defined in the PendTransUserMap.xml for the **<WindowsDomain>\<WindowsUserId>**. This command asks you to enter the password of the external user, UserID, used in this walkthrough.

```
ssomanage -setcredentials <WindowsDomain>\<WindowsUserId>  
WoodgroveBank.PendingTransactions
```

At the command prompt, run the following command to set the user credential for the Payment Tracker affiliate application. Use the **<DomainName>** and **<UserID>** defined in the PmntTrckUserMap.xml for the **<WindowsDomain>\<WindowsUserId>**. This command asks you to enter the password of the external user, PTUserID, used in this walkthrough.

```
ssomanage -setcredentials < WindowsDomain >\< WindowsUserId >  
WoodgroveBank.PaymentTracker
```

## **Deploy the BAM definition file for the Service Oriented Solution**

### **To deploy the BAM definition file for the Service Oriented Solution**

Open a command prompt, type the following command, and then press ENTER to set the path to find the BAM utilities:

```
SET PATH=%PATH%;%ProgramFiles%\ "Microsoft BizTalk Server 2006\Tracking"
```



At the command prompt, change the directory to the %BTSSolutionsPath%\SO\BTSSoln\BAM, type the following command, and then press ENTER:

**bm deploy-all -DefinitionFile:ServiceLevelTracking.xml**

## **Deploy the Service Oriented Solution**

### **To edit the binding files for the Service Oriented Solution**

At a command prompt, change the directory to the %BTSSolutionsPath%\SO\BTSSoln\Scripts folder, open the Deployallbinding.xml using Notepad, and then make the following edits:

Change the name of the server in the SET MGMT\_DB\_SERVER and MBMT\_DB to the name of the server and database that BizTalk Server is using.

Change the value of the SOLNDIR variable to "%BTSSolutionsPath%\SO\BTSSoln".

At a command prompt, change the directory to the %BTSSolutionsPath%\SO\BTSSoln\Bindings folder.

For the adapter version, open the AdapterSOAOrchBindings.xml using Notepad, and then edit as follows:

Replace all occurrences of \_\_MQ\_SERVER\_NAME\_\_ with the MQSeries Server name.

Replace all occurrences of \_\_MQ\_QMANAGER\_NAME\_\_ with the MQSeries Queue Manager name.

Replace all occurrences of \_\_PT\_WS\_SERVER\_NAME\_\_ in the string "<Address>https://\_\_PT\_WS\_SERVER\_NAME\_\_" with the server name where the Pending Transactions Web service is deployed. The server name must match the **common name** in the step, "To configure the Web server for SSL". You shouldn't use localhost.

For the inline version, open the InlineSOAOrchBindings.xml using Notepad, and then edit as follows:

Replace all occurrences of \_\_MQ\_SERVER\_NAME\_\_ with the MQSeries Server name.

Replace all occurrences of \_\_MQ\_QMANAGER\_NAME\_\_ with the MQSeries Queue Manager name.

### **To deploy the Service Oriented solution**

At a command prompt, change the directory to the %BTSSolutionsPath%\SO\BTSSoln\Scripts folder, type the following command, and then press ENTER.

Deployallbinding.cmd

Configure the stub Pending Transactions Web Services when a mainframe is not available

To configure the stub Pending Transactions Web service (for using the adapter version without a mainframe)

In the Internet Information Services (IIS) Manager, expand Web Sites, right-click the Default Web Site, point to New, and then click Virtual Directory to run Virtual Directory Creation Wizard.

Using the Virtual Directory Creation Wizard, create the following virtual directory for stub Pending Transactions Web service for the adapter version:

Alias = Microsoft.Samples.BizTalk.WoodgroveBank.StubPendingTransactions

PATH = <BizTalk Install  
Directory>\SDK\Scenarios\SO\BTSSoln\StubWebServices\PendingTrans

Access Permissions = Read, Run scripts

In the **Internet Information Services (IIS) Manager**, expand **Web Sites**, expand the **Default Web Site**, right-click Microsoft.Samples.BizTalk.WoodgroveBank.StubPendingTransactions, click **Properties**, and then modify the settings as follows using the **Properties** dialog box.

In the **Virtual directory** tab, set the **Application Pool** to <**YourAppPool**> you created in the step, "To create the virtual directories in IIS for the solution".

In the **Directory Security** tab, click **Edit** in the **Authentication and access control** group box, and then select **Enable anonymous access**. Click **OK** to exit.

In the **BizTalk Server Administration** console, expand **BizTalk Group**, expand **Applications**, expand BTSScn.SO.CustomerService, expand **Send Ports**, right-click **PendingTransactionSolicitResponsePort**, and then click **Properties**.

In the **General** page, click **Configure** to display the **Transport Properties** dialog box, and then modify the **Web Service URL** to the stub Pending Transaction Web service, for example:

http://localhost/Microsoft.Samples.BizTalk.WoodgroveBank.StubPendingTransactions/StubPendTransWS.asmx

Close all of the dialog boxes.

**To configure the stub Pending Transactions Web service (for using the inline version without a mainframe)**

In the **Internet Information Services (IIS) Manager**, expand **Web Sites**, right-click the **Default Web Site**, point to **New**, and then click **Virtual Directory** to run **Virtual Directory Creation Wizard**.

Using the **Virtual Directory Creation Wizard**, create the following virtual directory for stub Pending Transactions Web service for the adapter version:

Alias = Microsoft.Samples.BizTalk.WoodgroveBank.StubPendingTransactions

PATH = <BizTalk Install  
Directory>\SDK\Scenarios\SO\BTSSoln\StubWebServices\PendingTrans

Access Permissions = Read, Run scripts

In the **Internet Information Services (IIS) Manager**, expand **Web Sites**, expand the **Default Web Site**, right-click Microsoft.Samples.BizTalk.WoodgroveBank.StubPendingTransactions, click **Properties**, and then modify the settings as follows:

In the **Virtual directory** tab, set the **Application Pool** to <YourAppPool> you created in the step, "To create the virtual directories in IIS for the solution".

In the **Directory Security** tab, click **Edit** in the **Authentication and access control** group box, and then select **Enable anonymous access**. Click **OK** to exit.

Open a command prompt, and then change the directory to the %BTSSolutionsPath%\SO\BTSSoln\Scripts folder.

At the command prompt, open the SetConfigValuesInSSO.xml file using Notepad, and then set the value of the **PendingTransactionsInlineURL** to the URL of the stub Pending Transaction Web Service.

**http://localhost/Microsoft.Samples.BizTalk.WoodgroveBank.StubPendingTransactions/StubPendTransWS.asmx**

At the command prompt, type **SetConfigValuesInSSO.cmd**, and then press ENTER.

## Start the Service Oriented Solution

### To start the Service Oriented Solution

Open a command prompt, change the directory to the %BTSSolutionsPath%\SO\BTSSoln\Scripts folder, type the following command, and then press ENTER to start all orchestrations for the inline and adapter versions.

**startAll.vbs**

Run the service-oriented solution. For more information about running the solution.

## How to Run the Service Oriented Solution

The following steps describe how to run and validate the service oriented solution on a single computer. After starting the Payment Tracker simulator, you can send requests using either the

SOAP or MQSeries transport (with separate procedures for the adapter and inline versions of the service oriented solution).

Send requests by SOAP transport using the client application (stub version)

Send requests using the client application (adapter version)

Send requests using the client application (inline version)

### **Send requests by SOAP transport using the client application (stub version)**

#### **To send requests by SOAP transport using the client application (stub version)**

Open a command prompt, change the directory to the <BizTalk Server install Directory>\SDK\Scenarios\SO\BTSSoln\SimpleClient\bin\Debug, and then run the BTSScnSOSimpleClient.exe.

Type any characters in the **RequestType**, **RequestSource**, and **RequestID** text boxes.

Type any 16-digit number in the **Account Number** text box.

Select **SOAP (WS Call)** and **Stub** in the **Select Transport and Parameters** group box.

Type the following URL in the **URL** text box, for example:

**http://localhost/Microsoft.Samples.BizTalk.WoodgroveBank.OrchProxy.Stub/ CustomerServicePort.asmx**

Type **ZipCode** in the **Name** text box under **Authentication Elements**, and then type any characters in the **Value** text box.

Type **CustomerName** in the **Name** text box under **Authentication Elements**, and then type any characters in the **Value** text box.

Click **Get my balance**.

The response is displayed in the **Response** text box: **SUCCESS** appears if the request is handled successfully; an error message appears if the request fails.

### Send requests using the client application (adapter version)

#### To send requests using the client application (adapter version)

Open a command prompt, change the directory to <BizTalk Server install Directory>\SDK\Scenarios\SO\BTSSoln\PaymentTracker\bin\Debug and, then run the following command to start the PaymentTracker simulator:

```
BTSScnSOPaymentTracker.exe LastPaymentsInputQueue LastPaymentsOutputQueue <Queue Manager Name> 5 [<Channel Definition>]
```

Leave the Payment Tracker simulator running.

Open a command prompt, change the directory to the <BizTalk Server install Directory>\SDK\Scenarios\SO\BTSSoln\SimpleClient\bin\Debug, and then run the BTSScnSOSimpleClient.exe.

In the BTSScnSOSimpleClient.exe, send a request by SOAP transport using the as follows:

Type any characters in the **RequestType**, **RequestSource**, and **RequestID** text boxes.

Type any 16-digit number in the **Account Number** text box.

Select **SOAP (WS Call)** and **Adapter** in the **Select Transport and Parameters** group box.

Type the following URL in the **URL** text box, for example:

<http://localhost/Microsoft.Samples.BizTalk.WoodgroveBank.OrchProxy.Adapter/CustomServicePort.asmx>

Type **ZipCode** in the **Name** text box under **Authentication Elements**, and then type any characters in the **Value** text box.

Type **CustomerName** in the **Name** text box under **Authentication Elements**, and then type any characters in the **Value** text box.

Click **Get my balance**.

The response is displayed in the **Response** text box: **SUCCESS** appears if the request is handled successfully; an error message appears if the request fails.

In the BTSScnSOSimpleClient.exe, send requests by MQSeries transport as follows:

Type any characters in the **RequestType**, **RequestSource**, and **RequestID** text boxes.

Type a 16-digit number in the **Account Number** text box.

Select **MQSeries** in the **Select Transport and Parameters** group box.

Type **<Queue Manager Name>** in the **Queue Manager** text box. QM\_<Your Computer Name> is the default value for **<Queue Manager Name>**.

Type **AdapterSOAInputQueue** in the **Input Queue** text box.

Type **AdapterSOAOutputQueue** in the **Output Queue** text box.

Type **<Channel Definition>** in the **Channel Definition** box. S\_**<Your Computer Name>/TCP/<Your Computer Name>(1414)** is the default value for **<Channel Definition>**.

Type **ZipCode** in the **Name** text box under **Authentication Elements**, and then type any characters in the **Value** text box.

Type **CustomerName** in the **Name** text box under **Authentication Elements**, and then type any characters in the **Value** text box.

Click **Get my balance**.

The response is displayed in the **Response** text box: **SUCCESS** appears if the request is handled successfully; an error message appears if the request fails.

## Send requests using the client application (inline version)

### To send requests using the client application (inline version)

Open a command prompt, change the directory to **<BizTalk Server install Directory>\SDK\Scenarios\SO\BTSSoln\PaymentTracker\bin\Debug**, and then run the following command to start the PaymentTracker simulator:

```
BTSScnSOPaymentTracker.exe LastPaymentsInputQueue LastPaymentsOutputQueue <Queue Manager Name> 5 [<Channel Definition>]
```

Leave the Payment Tracker simulator running.

Open a command prompt, change the directory to the <BizTalk Server install Directory>\SDK\Scenarios\SO\BTSSoln\SimpleClient\bin\Debug, and then run the BTSScnSOSimpleClient.exe.

In the BTSScnSOSimpleClient.exe, send a request by SOAP transport using the as follows:

Type any characters in the **RequestType**, **RequestSource**, and **RequestID** text boxes.

Type any 16-digit number in the **Account Number** text box.

Select **SOAP (WS Call)** and **Inline** in the **Select Transport and Parameters** group box.

Type the following URL in the **URL** text box, for example:

**http://localhost/Microsoft.Samples.BizTalk.WoodgroveBank.OrchProxy.Inline/CustomerServicePort.asmx**

Type **ZipCode** in the **Name** text box under **Authentication Elements**, and then type any characters in the **Value** text box.

Type **CustomerName** in the **Name** text box under **Authentication Elements**, and then type any characters in the **Value** text box.

Click **Get my balance**.

The response is displayed in the **Response** text box: **SUCCESS** appears if the request is handled successfully; an error message appears if the request fails.

In the BTSScnSOSimpleClient.exe, send requests by MQSeries transport as follows:



Type any characters in the **RequestType**, **RequestSource**, and **RequestID** text boxes.

Type a 16-digit number in the **Account Number** text box.

Select **MQSeries** in the **Select Transport and Parameters** group box.

Type **<Queue Manager Name>** in the **Queue Manager** text box. QM\_<Your Computer Name> is the default value for **<Queue Manager Name>**.

Type **InlineSOAInputQueue** in the **Input Queue** text box.

Type **InlineSOAOutputQueue** in the **Output Queue** text box.

Type **<Channel Definition>** in the **Channel Definition** box. S\_<Your Computer Name>/TCP/<Your Computer Name>(1414) is the default value for **<Channel Definition>**.

Type **ZipCode** in the **Name** text box under **Authentication Elements**, and then type any characters in the **Value** text box.

Type **CustomerName** in the **Name** text box under **Authentication Elements**, and then type any characters in the **Value** text box.

Click **Get my balance**.

The response is displayed in the **Response** text box: **SUCCESS** appears if the request is handled successfully; an error message appears if the request fails.

The screenshot shows the 'Service Oriented Scenario - Simple Client' application window. It is divided into several sections:

- Request:** Contains text boxes for 'Request Type', 'Request Source', 'Request ID' (with value '123456'), and 'Account Number' (with value '1234567890120002').
- Authentication Elements:** A table with two columns: 'Name' and 'Value'.
 

Name	Value
ZipCode	90052
CustomerName	Woodgrove
- Get my balance!**: A button located below the authentication elements.
- Select Transport and Parameters:** A section with radio buttons for 'SOAP (WS Call)' and 'MQSeries' (which is selected). Below it are 'Web Service Details' including a 'URL' field with the value 'http://localhost:10000/Online/CustomServicePort.asmx'.
  - MQSeries Transport Parameters:**
    - Queue Manager: QM\_bpi2x\_015
    - Input Queue: InlineSOAInputQueue
    - Output Queue: InlineSOAOutputQueue
    - Channel Definition: SYSTEM.DEF.SVRCONN/TCP/bpi2x\_015!channel-name/transport/hostname(port)
- Response:** A section displaying the results of the request.
 

Credit Limit: 10000	Account Balance: 36032.0	Last Payment: 26644
Pending Trans: 611.2	Cash Advance Limit: 1000	Time taken: 659.3805 Milliseconds
- SUCCESS:** A large text area at the bottom showing the word 'SUCCESS'.

## Deploying the SO Solution to Production Servers

### In This Section

- Prerequisites of the SO Solution Production Servers
- How to Prepare the SO Solution for Deployment
- How to Prepare the Production Servers for the SO Solution
- How to Install the SO Solution to Production Servers
- How to Run the SO Solution on Production Servers
- How to Remove a Version of the SO Solution

## Prerequisites of the SO Solution Production Servers

The service oriented solution can consist of different computers, and each computer plays an integral part in the overall solution. The hardware and software prerequisites for each computer are as follows: .

Domain Controller computer

Microsoft Windows Server 2003 Enterprise Edition

Two BizTalk Server computers

Windows Server 2003 Enterprise Edition

Microsoft BizTalk Server 2006

Two Network Interface Cards (NIC) per server for Network Load Balancing (NLB)

IBM WebSphere MQ, client components for Windows platforms for the Inline version.

Two Microsoft SQL Server Cluster nodes for the BizTalk Server Configuration and Message Box databases.

Windows Server 2003 Enterprise Edition

SQL Server 2000 with Service Pack 4

Two NICs per SQL Cluster node

A shared storage resource

Administration Client computer for managing the BizTalk Server computers

- Windows Server 2003 Enterprise Edition

- BizTalk Server 2006

Application Client computer

- Windows Server 2003 Enterprise Edition

- Microsoft .NET Framework 2.0

External legacy system

- Windows Server 2003 Enterprise Edition

- IBM WebSphere MQ

- BizTalk Server 2006 MQ Adapter component

- Microsoft Host Integration Server 2004

- Microsoft .NET Framework 2.0

- SQL Server 2000 with Service Pack 4

## How to Prepare the SO Solution for Deployment

The following steps explain how to prepare the resources for the service-oriented solution for deployment to the production servers.

- To prepare resources for deployment to the BizTalk servers

- To prepare resources to be used from the Administration Client

- To prepare resources for the Internal Client

- To prepare resources for the Legacy System

### Prerequisites

Before performing the procedures in this topic, you must complete the steps in How to Install the Inline and Adapter Versions of the Service Oriented Solution.

### To prepare resources for deployment to BizTalk Servers

- Compile the service-oriented solution for a release configuration:

Search for and terminate all the service instances of the service oriented solution. For more information, see [How to Search for All Service Instances](#) and [How to Suspend, Resume, and Terminate Orchestration Instances](#).

Stop the host instance that is running the service-oriented solution. For more information, see [How to Stop a Host Instance](#).

At the command prompt, type the following command to set the %BTSSolutionsPath% environment variable to indicate the base folder of the E2E solutions:

```
setx BTSSolutionsPath %ProgramFiles%\Microsoft BizTalk Server 2006\SDK\Scenarios
```

At the command prompt, change the directory to the %BTSSolutionPath%\SO\BTSSoln\OrchProxy\Adapter folder, open the Web.config file using Notepad, and then modify the debug option as follows:

Replace debug="true" with debug="false".

Delete compilerOptions="/d:DEBUG;TRACE".

At the command prompt, change the directory to the %BTSSolutionPath%\SO\BTSSoln\OrchProxy\Inline folder, open the Web.config file using Notepad, and then modify the debug option as follows:

Replace debug="true" with debug="false".

Delete compilerOptions="/d:DEBUG;TRACE".

At the command prompt, change the directory to the %BTSSolutionPath%\SO\BTSSoln\StubWebServices\SAP folder, open the Web.config file using Notepad, and then modify the debug option as follows:

Replace debug="true" with debug="false".

Delete compilerOptions="/d:DEBUG;TRACE".

At the command prompt, change the directory to the %BTSSolutionPath%\SO\MFAccess\PendingTransactions folder, open the Web.config file using Notepad, and then modify the debug option as follows:

Replace debug="true" with debug="false".

Click Start, point to All Programs, point to Microsoft Visual Studio .NET 2003, point to Visual Studio .NET Tools, and then click Visual Studio .NET 2003 Command Prompt.

At the Visual Studio .NET 2003 Command Prompt, change the current folder to %BTSSolutionsPath%\SO\MFAccess, and then run the following command to compile the file Microsoft.Samples.BizTalk.WoodgroveBank.MainframeAccess.sln:

```
devEnv Microsoft.Samples.BizTalk.WoodgroveBank.MainframeAccess.sln /build Release /Out Build.log
```

Click Start, point to All Programs, point to Microsoft Visual Studio 2005, point to Visual Studio Tools, and then click Visual Studio 2005 Command Prompt.

At the Visual Studio 2005 Command Prompt, change the directory to the %BTSolutionsPath%\Common\SSOApplicationConfig folder, and then run the following command to compile the SSOApplicationConfig.sln file.

```
devenv SSOApplicationConfig.sln /Build "Release|AnyCPU" /Out Build.log
```

At the Visual Studio 2005 Command Prompt, change the directory to the %BTSolutionsPath%\SO\BTSSoln folder, and then run the following commands to compile and deploy the service-oriented solution.

```
devenv Microsoft.Samples.BizTalk.WoodgroveBank.sln /Build "Release|Mixed Platforms" /Out Build.log
```

```
devenv Microsoft.Samples.BizTalk.WoodgroveBank.sln /Deploy "Release|Mixed Platforms" /Out Build.log
```

```
btstask AddResource -ApplicationName: BTSScn.SO.CustomerService -Type: Assembly -  
OverWrite -Options: GacOnAdd,GacOnInstall -  
Source: ConfigHelper\bin\Release\Microsoft.Samples.BizTalk.WoodgroveBank.ConfigHelp  
er.dll
```

```
btstask AddResource -ApplicationName: BTSScn.SO.CustomerService -Type: Assembly -  
OverWrite -Options: GacOnAdd,GacOnInstall -  
Source: PaymentTrackerCall\bin\Release\Microsoft.Samples.BizTalk.WoodgroveBank.Pay  
mentTrackerCall.dll
```

```
btstask AddResource -ApplicationName: BTSScn.SO.CustomerService -Type: Assembly -  
OverWrite -Options: GacOnAdd,GacOnInstall -  
Source: PendTransCall\bin\Release\Microsoft.Samples.BizTalk.WoodgroveBank.PendingT  
ransactionsCall.dll
```

```
btstask AddResource -ApplicationName: BTSScn.SO.CustomerService -Type: Assembly -  
OverWrite -Options: GacOnAdd,GacOnInstall -  
Source: StubWebServices\StubSAPCall\bin\Release\Microsoft.Samples.BizTalk.Woodgro  
veBank.StubSAPCall.dll
```

```
btstask AddResource -ApplicationName: BTSScn.SO.CustomerService -Type: Assembly -  
OverWrite -Options: GacOnAdd,GacOnInstall -  
Source: SchemaClasses\bin\Release\Microsoft.Samples.BizTalk.WoodgroveBank.Schema  
Classes.dll
```

```
btstask AddResource -ApplicationName: BTSScn.SO.CustomerService -Type: Assembly -  
OverWrite -Options: GacOnAdd,GacOnInstall -  
Source: ServiceLevelTracking\bin\Release\Microsoft.Samples.BizTalk.WoodgroveBank.Se  
rviceLevelTracking.dll
```

```
btstask AddResource -ApplicationName: BTSScn.SO.CustomerService -Type: Assembly -  
OverWrite -Options: GacOnAdd,GacOnInstall -  
Source: ErrorHandler\bin\Release\Microsoft.Samples.BizTalk.WoodgroveBank.ErrorHelp  
er.dll
```

```
btstask AddResource -ApplicationName: BTSScn.SO.CustomerService -Type: Assembly -  
OverWrite -Options: GacOnAdd,GacOnInstall -  
Source: Utilities\bin\Release\Microsoft.Samples.BizTalk.WoodgroveBank.Utilities.dll
```

```
btstask AddResource -ApplicationName: BTSScn.SO.CustomerService -Type: Assembly -  
OverWrite -Options: GacOnAdd,GacOnInstall -  
Source: InPipelineComp\bin\Release\Microsoft.Samples.BizTalk.WoodgroveBank.InputPi  
pipelineComponents.dll
```

```
btstask AddResource -ApplicationName: BTSScn.SO.CustomerService -Type: Assembly -  
OverWrite -Options: GacOnAdd,GacOnInstall -  
Source: PmTrkPipelineComp\bin\Release\Microsoft.Samples.BizTalk.WoodgroveBank.Pay  
mentTrackerPipelineComponents.dll
```

In the C:\ folder, create the folder BizTalkServer.

Open the BizTalk Server Administration console as follows: click Start, point to All Programs, point to BizTalk Server 2006, and then click BizTalk Server Administration.

In the BizTalk Server Administration console, expand BizTalk Server 2006 Administration, expand BizTalk Group, expand Applications, right-click BTSScn.SO.CustomerService, point to Export, and then click MSI files.

On the Welcome page, click Next.

On the Select Resources page, clear the Bindings check box, and then click Next.

On the Specify IIS Hosts page, keep all of the default settings, and then click Next.

On the Dependencies page, click Next.

On the Destination page, type C:\deploy\BizTalkServer\BTSScn.SO.CustomerService.msi in the MSI file to generate box, and then click Export.

On the Summary page, click Finish.

To prepare resources to be used from the Administration Client

In the C:\deploy folder, create the folder AdminClient.

If you installed the stub version of the solution, remove it. If you don't remove the stub version, you must delete all references to the stub version from the binding file that you will generate in the next step because you won't install the stub version on the production servers. For more information about how to remove the stub version, see the step "To remove the stub version of the Service Oriented solution" in How to Install the Inline and Adapter Versions of the Service Oriented Solution.

Create a binding file for the BTSScn.SO.CustomerService application.

In the BizTalk Server Administration console, expand BizTalk Server 2006 Administration, expand BizTalk Group, expand Applications, right-click BTSScn.SO.CustomerService, point to Export, and then click Binding.

On the Export Bindings page, select the Export all bindings from the current application and Export Global Party Information check boxes.

In the Export to file name box type  
c:\deploy\AdminClient\BTSScn.SO.CustomerService.BindingInfo.xml, and then click OK.

At a command prompt, change the directory to the %BTSSolutionsPath%\SO\BTSSoln\BAM folder, and then copy ServiceLevel.xml to the AdminClient folder.

At a command prompt, change the directory to the %BTSSolutionsPath%\SO\BTSSoln\Scripts folder, and then copy the following files. They will be used for SSO configuration.

ConfigStoreApp.xml

CreateInitialConfigInSSO.cmd

PendTransAffApp.xml

PendTransUserMap.xml

PmntTrckAffApp.xml

PmntTrckUserMap.xml

SetConfigValuesInSSO.cmd

At a command prompt, change the directory to the  
%BTSSolutionsPath%\Common\SSOApplicationConfig\bin folder, and then copy the  
BTSScnSSOApplicationConfig.exe file in the AdminClient folder.

Copy the MSI files in the BizTalkServer folder that you created in the previous step.

### **To prepare resources for the Application Client**

In the C:\deploy folder, create the folder ApplicationClient.

At a command prompt, change the directory to the  
%BTSSolutionsPath%\SO\BTSSoln\SimpleClient\bin\Release folder, and then copy all the  
files in the folder to the ApplicationClient folder.

### **To prepare the resource for the Legacy System**

Create the following folders:

C:\deploy\SOLegacy

C:\deploy\SOLegacy\HISTIComponent

C:\deploy\SOLegacy\HISTISimpleTest

C:\deploy\SOLegacy\PaymentTracker

At a command prompt, change the directory to the %BTSSolutionsPath%\SO\MFAccess\HISTIComponent folder, and then copy the SOHISTIUsingCOM.TLB file to the folder C:\deploy\SOLegacy\HISTIComponent.

At a command prompt, change the directory to the %BTSSolutionsPath%\SO\MFAccess\HISTISimpleTest\bin\Release folder, and then copy all of the executable files into the C:\deploy\SOLegacy\HISTISimpleTest folder.

At a command prompt, change the directory to the %BTSSolutionsPath%\SO\MFAccess\HISTISimpleTest folder, and then copy the file Interop.SOHISTIUsingCOM.dll.reg to the C:\deploy\SOLegacy\HISTISimpleTest folder.

At a command prompt, change the directory to the %BTSSolutionsPath%\SO\BTSSoln\PaymentTracker\bin\Release folder, and then copy all the executable files into the C:\deploy\SOLegacy\PaymentTracker folder.

At a command prompt, change the directory to the %BTSSolutionsPath%\SO\MFAccess folder, and then copy the PendingTransactions folder into the C:\deploy\SOLegacy folder.

At a command prompt, change the directory to the %BTSSolutionsPath%\SO\BTSSoln folder, and then copy the SAP folder into the C:\deploy\SOLegacy folder.

## How to Prepare the Production Servers for the SO Solution

The following steps describe how to prepare the production servers for the service oriented solution:

To create a domain controller

To create global domain security groups and user accounts to be used by the service oriented solution

To configure SQL Cluster

To install the first BizTalk Server computer

To configure the first BizTalk Server computer

To install the second BizTalk Server computer

To configure the second BizTalk Server computer



To set up TCP/IP for configuring Network Load Balancing (NLB) between the BizTalk Server computers

To install IBM WebSphere MQ, client components for Windows platforms to the both BizTalk Server computers

To prepare the Administration Client computer

To configure the Administration Client computer

To prepare the Application Client computer

To prepare the Legacy System

The following table lists the server names used in this topic, and the names of the applications that you will deploy.

Network name	Applications/Services	Description
E2E-BTS1	BTSScn.SO.CustomerService Master Secret Service	BizTalk Server
E2E-BTS2	BTSScn.SO.CustomerService	BizTalk Server
E2E-SQL1	BizTalk Server databases SSO database	SQL Server
E2E-SQL2	BizTalk Server databases SSO database	SQL Server
E2E-ADMIN	BizTalk Administration tools	Administration client
E2E-CLIENT	SimpleClient.exe	Application client
E2E-LEGACY	IBM WebSphere MQSeries Server AdpaterSOAInputQueue AdapterSOAOutputQueue InlineSOAInputQueue InlineSOAOutputQueue LastPaymentsInputQueue	Legacy system

	LastPaymentsOutputQueue Microsoft Host Integration Server 2004 BTSScn SO TI Component MQSeries Agent PaymentTracker.exe PendingTransactions Web application Stub SAP WebServices	
E2E-DC	Domain controller	Domain controller

## Prerequisites

Before performing the steps in this topic, you must complete the steps in Developer Machine Setup for the SO Solution

## To create a domain controller

For this procedure, use E2E-DC for the name of your domain controller.

Log on to the E2E-DC domain controller using the local Administrator account.

Click Start, click Run, type dcpromo, and then press ENTER. The Active Directory Installation Wizard is started.

On the Domain Controller Type page, select the Domain controller for a new domain check box, and then click Next.

On the Create New Domain page, select the Domain in a new forest check box, and then click Next.

On the New Domain Name page, type contoso.com in the Full DNS name for the new domain box, and then click Next.

On the NetBIOS Domain Name page, type CONTOSO in the Domain NetBIOS name box, and then click Next.

On the Database and Log Folders page, click Next.

On the Shared System Volume page, click Next.

On the DNS Registration Diagnostics page, select the Install and configure the DNS server on this computer, and set this computer to use this DNS server as its preferred DNS server check box, and then click Next.

On the Permissions page, select the Permissions compatible only with Windows 2000 or Windows Server 2003 operating systems check box, and then click Next.

On the Directory Services Restore Mode Administrator Password page, type and confirm the password you want to assign to this server's Administrator account, and then click Next. This is the account that will be used when the computer is started in Directory Services Restore Mode.

On the Summary page, click Next to begin installing Active Directory.

After Active Directory is installed, restart the computer.

To create global domain security groups and user accounts to be used by the service oriented solution

Log on to the E2E-DC domain controller using the Domain Administrator account.

Click Start, point to All Programs, point to Administrative Tools, and then click Active Directory Users and Computers to start the Active Directory Users and Computers console.

In the Active Directory Users and Computers console, create the following new organizational units in the contoso.com domain:

BTSAccounts

BTSAccounts\btsntsvc

BTSAccounts\Groups

BTSAccounts\Groups\Global

BTSAccounts\Users

In the Active Directory Users and Computers console, create the following new domain groups under the BTSAccounts\Groups\Global organizational unit:

SSO Administrators

SSO Affiliate Administrators

BizTalk Server Administrators

BizTalk Server Operators

BizTalk MQSeries Adapter Host Users

BizTalk SO Adapter Version Host Users

BizTalk SO Inline Version Host Users

BizTalk SOAP Trusted Host Users

BizTalk SOAP Isolated Trusted Host Users

BizTalk Tracking Host Users

Customer Service Users

Payment Tracker Users

Pending Transactions Users

SQL Service Group

In the **Active Directory Users and Computers** console, create the following domain users under the BTSAccounts\Users organizational unit, and add them to the appropriate domain groups. These are domain groups you created in the previous step.

Domain user name	Domain name description	Members of
ssoadmin	SSO Administrator	SSO Administrators
btsadmin	BizTalk Administrator	SSO Affiliate Administrators BizTalk Server Administrators SSO Administrators
btsinstall	BizTalk Installation	SSO Administrators
ssoslave1	SSO Service	SSO Administrators
ssomaster	SSO Master Secret	SSO Administrators
sqlservice	SQL Service	SQL Service Group
btsoperator	BizTalk Server Operators	BizTalk Server Operators
User1	Simple Client user account	Payment Tracker Users Pending Transactions Users

In the **Active Directory Users and Computers** console, create domain users under the BTSAccounts\btsntsvc organizational unit, as shown in the following table. Add them to the

groups you created in Step 4. When you create the domain users, select the **User cannot change password** check box. Verify that the **Password never expires** and **User must change password at next logon** check boxes are clear.

Domain user name	Domain name description	Members of
btssoinline	BizTalk SO Inline Version	BizTalk SO Inline Version Host Users Customer Service Users SSO Administrators
btssoadapter	BizTalk SO Adapter Version	BizTalk SO Adapter Version Host Users Customer Service Users
btssoapistrusted	BizTalk SOAP Adapter Isolated	BizTalk SOAP Isolated Trusted Host Users
btssoaptrusted	BizTalk SOAP Adapter	BizTalk SOAP Trusted Host Users SSO Administrators
btsibmmqadapter	BizTalk MQSeries Adapter	BizTalk MQSeries Adapter Host Users SSO Administrators
btstracking	BizTalk Tracking	BizTalk Tracking Host Users

In the **Active Directory Users and Computers** console, add the BizTalk service account, btsibmmqadapter, for the MQSeries Adapter to the Pending Transactions Users and Payment Tracker Users domain group. You don't need to add btsibmmqadapter if you will use the client application with only the SOAP transport.

### To configure a SQL Cluster

For this procedure, use E2E-SQL1 and E2E-SQL2 for the computer names of the cluster nodes.

Join all cluster nodes to the contoso.com domain.

Install and configure Windows Server 2003 Clustering Services on the cluster nodes.

Configure Microsoft Distributed Transaction Coordinator (MS DTC) on one of the cluster nodes.

Verify that Network DTC Access is enabled.

Install and configure the SQL cluster on one of the cluster nodes.

For this procedure, use E2E-SQL for the name of the SQL virtual server.

Choose the CONTOSO\sqlservice domain user account for SQL Server and the SQL Server Agent service.

Select the **Windows Authentication Mode** check box.

Verify that the SQL Server service is configured for auto-start.

Install SQL Server 2000 Service Pack 4 on the one of the cluster nodes.

On both the E2E-SQL1 computer and the E2E-SQL2 computer, add the CONTOSO\btsinstall and CONTOSO\btsadmin accounts to the local Administrators groups.

### **To install the first BizTalk Server computer**

For this procedure, use E2E-BTS1 for the computer name of the first BizTalk Server.

Log on to E2E-BTS1 using the local Administrator account.

Join the E2E-BTS1 computer to the contoso.com domain.

Click **Start**, point to **All Programs**, point to **Administrative Tools**, click **Computer Management**, and then add CONTOSO\btsinstall and CONTOSO\btsadmin accounts to the local Administrators group.

Restart E2E-BTS1.

Log on to E2E-BTS1 using the CONTOSO\btsinstall domain account.

Copy the Redistributable CAB (BtsRedistW2k3EN32.cab) file to the local computer.

Install ASP.NET and Internet Information Services:

Click **Start**, point to **Control Panel**, click **Add or Remove Programs**, and then click **Add/Remove Windows Components**.

On the **Windows Component Wizard**, double-click **Application Server**.

On the **Application Server** dialog box, select **ASP.NET** and **Internet Information Services (IIS)**, and then click **OK** to close the dialog box.

On the **Windows Component Wizard**, click **Next**.

Verify that Network DTC Access is enabled:

Click **Start**, point to **Control Panel**, click **Add or Remove Programs**, and then click **Add/Remove Windows Components**.

On the **Windows Component Wizard** page, double-click **Application Server**.

In the **Application Server** dialog box, select **Enable network DTC access**, and then click **OK** to close the dialog box.

Close all of the dialog boxes.

At a command prompt, type the following command, and then press ENTER to open the  
Component Services console:

**%SystemRoot%\system32\com\comexp.msc**

In the **Component Services** console, expand **Component Services**, expand **Computers**, right-click **My Computer**, and then click **Properties**.

In the **My Computer Properties** dialog box, click **MSDTC** tab, and then click **Security Configuration** on the **MSDTC** tab.

In the Security Configuration dialog box, select the following options:

Network DTC Access

Allow Remote Clients

Allow Remote Administrations

Allow Inbound

Allow Outbound

Enable XA Transactions

Insert the Microsoft BizTalk Server 2006 installation disk into the CD-ROM drive. If the BizTalk Server Setup page does not appear, double-click **setup.exe**, located in the root of the installation CD.

On the Microsoft BizTalk Server 2006 Installation Wizard page, click Install Microsoft BizTalk Server 2006 on this computer.

On the Customer Information page, type your user name and organization, and then click Next.

On the License Agreement page, click the Yes, I accept the terms of the license agreement check box to accept the terms of the License Agreement, and then click Next.

On the Component Installation page, select the following components to install:

Server Runtime

Administration Tools

Enterprise Single Sign-On Administration

Enterprise Single Sign-On Master Secret Server

BAM Event API

On the Redistributable Prerequisites page, select the Automatically install the redistributable prerequisites from a CAB file check box, click Browse, and then choose the BtsRedistW2k3EN32.cab file, which you copied in the previous step.

To configure the first BizTalk Server computer

Log on to E2E-BTS1 using CONTOSO\btsinstall domain account.

At the command prompt, change the directory to the <BizTalk Install Directory> folder, type configuration.exe, and then press ENTER.

On the Microsoft BizTalk Server 2006 Configuration page - Start page, perform the following steps:

Select the Custom configuration check box.

Type E2E-SQL in the Database server name box.

Click Configure.

On the Enterprise Single Sign-On page, perform the following steps:

Select the Enable Enterprise Single Sign-On on this computer check box.

Select the Create a new SSO system check box.

Choose Windows service for the SSO service listed in the following table:

Name	Account
<b>Enterprise Single Sign-On Service</b>	CONTOSO\ssomaster

Choose **Windows accounts** for the SSO roles listed in the listed the following table:

Role Name	Windows Group
<b>SSO Administrator(s)</b>	CONTOSO\SSO Administrator
<b>SSO Affiliate Administrator(s)</b>	CONTOSO\SSO Affiliate Administrators

On the Enterprise Single Sign-On Secret Backup page, type the Secret backup password, Confirm password, and Password reminder, and then click OK.

On the Group page, perform the following steps:



Select the Enable BizTalk Server Group on this computer check box.

Select the Create a new BizTalk Group check box.

Choose BizTalk Administrative Roles as the following table:

Role Name	Windows Group
<b>BizTalk Administrators Group</b>	CONTOSO\BizTalk Server Administrators
<b>BizTalk Operators Group</b>	CONTOSO\BizTalk Server Operators

On the BizTalk Server Runtime page, perform the following steps:

Select the Register the BizTalk Server runtime components check box.

Clear the Create BizTalk In-Process Host and Host Instance check box.

Clear the Create BizTalk Isolated Host and Host Instance check box.

On the tool bar of the Microsoft BizTalk Server 2006 Configuration, click Apply Configuration, click Configure on the Microsoft BizTalk Server 2006 Configuration Wizard page, and then click Finish.

In the Computer Management console, add the CONTOSO\BizTalk SOAP Adapter Isolated domain group to the local IIS\_WPG group.

To install the second BizTalk Server computer

For this procedure, use E2E-BTS2 for the computer name of the second BizTalk Server.

Log on to E2E-BTS2 using the local Administrator account.

Join E2E-BTS2 to the contoso.com domain.

In the Computer Management console, add the CONTOSO\btsinstall and CONTOSO\btsadmin accounts to the local Administrators group.

Restart the E2E-BTS2 computer.

Log on to E2E-BTS2 using CONTOSO\btsinstall domain account.

Copy the Redistributable CAB (BtsRedistW2k3EN32.cab) file to the local computer.

Install ASP.NET and Internet Information Services:

Click Start, point to Control Panel, click Add or Remove Programs, and then click Add/Remove Windows Components.

On the Windows Component Wizard page, double-click Application Server.

In the Application Server dialog box, select ASP.NET and Internet Information Services (IIS), and then click OK to close the dialog box.

On the Windows Component Wizard page, click Next.

Verify that Network DTC Access is enabled:

Click Start, point to Control Panel, click Add or Remove Programs, and then click Add/Remove Windows Components.

On the Windows Component Wizard, double-click Application Server.

On the Application Server dialog box, select the Enable network DTC access check box, and then click OK to close the dialog box.

Close all of the dialog boxes.

At a command prompt, type the following command, and then press ENTER to open the Component Services console:

```
%SystemRoot%\system32\com\comexp.msc
```

In the Component Services console, expand Component Services, expand Computers, right-click My Computer, and then click Properties.

In the My Computer Properties dialog box, click the MSDTC tab, and then click Security Configuration in the MSDTC tab.

In the Security Configuration dialog box, select the following options:

Network DTC Access

Allow Remote Clients

Allow Remote Administrations

Allow Inbound

Allow Outbound

Enable XA Transactions.

Insert the Microsoft BizTalk Server 2006 installation disk into the CD-ROM drive. If the BizTalk Server Setup page does not appear, double-click setup.exe located in the root of the installation CD.

On the Microsoft BizTalk Server 2006 Installation Wizard page, click Install Microsoft BizTalk Server 2006 on this computer.

On the Customer Information page, type your user name and organization, and then click Next.

On the License Agreement page, click Yes, I accept the terms of the license agreement to accept the terms of the License Agreement, and then click Next.

On the Component Installation page, select the following components to install:

Server Runtime

BAM Event API

On the Redistributable Prerequisites page, select the Automatically install the redistributable prerequisites from a CAB file check box, click Browse, and then choose the BtsRedistW2k3EN32.cab file which you copied in the previous step.

To configure the second BizTalk Server computer

Log on to E2E-BTS2 using CONTOSO\btsinstall domain account.

At a command prompt, change the directory to the <BizTalk Install Directory> folder, type configuration.exe, and then press ENTER.

On the Microsoft BizTalk Server 2006 Configuration page - Start page, perform the following steps:

Select the Custom configuration check box.

Type E2E-SQL in the Database server name box.

Click Configure.

On the Enterprise Single Sign-On page, perform the following steps:

Select the Enable Enterprise Single Sign-On on this computer check box.

Select the Join an existing SSO System check box.

Choose Windows service for the SSO service listed in the following table:

Name	Account
Enterprise Single Sign-On Service	CONTOSO\ssoslave1

On the Group page, perform the following steps:

Select the Enable BizTalk Server Group on this computer check box.

Select the Join an existing BizTalk Group check box.

On the BizTalk Server Runtime page, perform the following steps:

- Select the Register the BizTalk Server runtime components check box.

- Clear the Create BizTalk In-Process Host and Host Instance check box.

- Clear the Create BizTalk Isolated Host and Host Instance check box.

On the tool bar of the Microsoft BizTalk Server 2006 Configuration, click Apply Configuration, click Configure on the Microsoft BizTalk Server 2006 Configuration Wizard page, and then click Finish.

In the Computer Management console, add the CONTOSO\BizTalk SOAP Adapter Isolated domain group to the local IIS\_WPG group.

To setup TCP/IP for configuring Network Load Balancing (NLB) between the BizTalk Server computers

Configure the Network Load Balancing and TCP/IP components on the first BizTalk Server computer.

For this procedure, use E2E-BTS.contoso.com for the full Internet name.

Configure the Network Load Balancing and TCP/IP components on the second BizTalk Server computer.

Add a new host name, E2E-BTS, to the DNS on the E2E-DC domain controller.

To install IBM WebSphere MQ, client components for Windows platforms to the both BizTalk Server computers

Install the IBM WebSphere MQ client components for Windows platform. When you install these components, be sure to do the following:

- Select Custom Select as the Setup Type.

- Install the Windows Client component.

- Keep all of the default settings.

- If you are using the IBM WebSphere MQ for Windows platforms v5.3, install the Fix Pack 10 (CSD10).

To prepare the Administration Client computer

For this procedure, use E2E-ADMIN for the computer name of the administration client.

Log on to E2E-ADMIN using the local Administrator account.

Join the E2E-ADMIN computer to the contoso.com domain.

In the Computer Management console, add the CONTOSO\btsinstall account and the CONTOSO\btsadmin account to the local Administrators group.

Restart the E2E-ADMIN computer.

Log on to E2E-ADMIN using CONTOSO\btsinstall domain account.

Copy the Redistributable CAB (BtsRedistW2k3EN32.cab) file to the local computer.

Verify that Network DTC Access is enabled:

Click Start, point to Control Panel, click Add or Remove Programs, and then click Add/Remove Windows Components.

On the Windows Component Wizard page, double-click Application Server.

In the Application Server dialog box, select Enable network DTC access, and then click OK to close the dialog box.

Close all of the dialog boxes.

At a command prompt, type the following command, and then press ENTER to open the Component Services console:

```
%SystemRoot%\system32\com\comexp.msc
```

In the Component Services console, expand Component Services, expand Computers, right-click My Computer, and then click Properties.

In the My Computer Properties dialog box, click MSDTC tab, and then click Security Configuration in the MSDTC tab.

In the Security Configuration dialog box, check the following options box:

Network DTC Access

Allow Remote Clients

Allow Remote Administrations

Allow Inbound

Allow Outbound

Enable XA Transactions.

Insert the Microsoft BizTalk Server 2006 installation disk into the CD-ROM drive. If the BizTalk Server Setup page does not appear, double-click setup.exe located in the root of the installation CD.

On the Microsoft BizTalk Server 2006 Installation Wizard page, click Install Microsoft BizTalk Server 2006 on this computer.

On the Customer Information page, type your user name and organization, and then click Next.

On the License Agreement page, click the Yes, I accept the terms of the license agreement check box to accept the terms of the License Agreement, and then click Next.

On the Component Installation page, select the following components to install:

Documentation

Administration

Tools

Enterprise Single Sign-On Administration

On the Redistributable Prerequisites page, select the Automatically install the redistributable prerequisites from a CAB file check box, click Browse, and then choose the BtsRedistW2k3EN32.cab file which you copied in the previous step.

Run the SQL Server 2000 setup program.

On the Installation Definition dialog box, be sure to select the Client tools only check box.

Install SQL Server 2000 Service Pack 4.

To configure the Administration Client computer

Log on to E2E-ADMIN using CONTOSO\btsinstall domain account.

At a command prompt, change the directory to the <BizTalk Install Directory> folder, type configuration.exe, and then press ENTER.

On the Microsoft BizTalk Server 2006 Configuration page - Start page, perform the following steps:

Select the Custom configuration check box.

Type E2E-SQL in the Database server name box.

Click Configure.

On the Group page, perform the following steps:

Select the Enable BizTalk Server Group on this computer check box.

Select the Join an existing BizTalk Group check box.

On the tool bar of the Microsoft BizTalk Server 2006 Configuration, click Apply Configuration, click Configure on the Microsoft BizTalk Server 2006 Configuration Wizard page, and then click Finish.

To prepare the Application Client computer

For this procedure, use E2E-CLIENT for the computer name of the application client.

Log on to E2E-CLIENT using the local Administrator account.

Join the E2E-CLIENT computer to the contoso.com domain.

Restart the E2E-CLIENT computer.

Log on to E2E-CLIENT using the local Administrator account.

Install the IBM WebSphere MQ client components for Windows platform. . When you install these components, be sure to do the following:

Select Custom Select as the Setup Type.

Install the Windows Client component.

Keep all of the default settings.

If you are using the IBM WebSphere MQ for Windows platforms v5.3, install the Fix Pack 10 (CSD10). ☐ To prepare the Legacy System

For this procedure, use E2E-LEGACY for the computer name of the legacy system.

Make sure that you do not join the legacy system to contoso.com domain.

Log on to E2E-LEGACY using the local Administrator account.

Copy the Redistributable CAB (BtsRedistW2k3EN32.cab) file to the local computer.

Install ASP.NET and Internet Information Services:

Click Start, point to Control Panel, click Add or Remove Programs, and then click Add/Remove Windows Components.

On the Windows Component Wizard page, double-click Application Server.

On the Application Server dialog box, select ASP.NET and Internet Information Services (IIS), and then click OK to close the dialog box.

On the Windows Component Wizard page, click Next.

Install SQL Server 2000 and Service Pack 4.

Make sure that Network DTC Access is enabled:

Click Start, point to Control Panel, click Add or Remove Programs, and then click Add/Remove Windows Components.

On the Windows Component Wizard page, double-click Application Server.

In the Application Server dialog box, select the Enable network DTC access check box, and then click OK to close the dialog box.

Close all of the dialog boxes.

At a command prompt, type the following command, and then press ENTER to open the Component Services console:

```
%SystemRoot%\system32\com\comexp.msc
```

In the Component Services console, expand Component Services, expand Computers, right-click My Computer, and then click Properties.

In the My Computer Properties dialog box, click the MSDTC tab, and then click Security Configuration in the MSDTC tab.

In the Security Configuration dialog box, select the following options:

Network DTC Access

Allow Remote Clients

Allow Remote Administrations

Allow Inbound

Allow Outbound

Enable XA Transactions.

In the Computer Management console, create the btsibmmqadapter and btssoinline local user accounts for MQSeries Adapter and the inline version. The passwords and names must be the same as the ones used in the step "To create global domain security groups and user accounts used by the service oriented solution". Add btsibmmqadapter to the Distributed COM Users local group.

In the Computer Management console, create the sapappool and pendingtranapppool local user accounts that will be used for IIS application pools. Then, add them to the local IIS\_WPG group.

In the Computer Management console, create the btssoticomsvc and mqagentsvc local user accounts under which COM+ applications will run.



In the Computer Management console, create the hissvc local user account for HIS LU62 Resync service account.

In the Computer Management console, create the PTUser1 and PTUser2 user accounts to represent the users of the SAP and Pending Transactions applications.

In the Computer Management console, create the User1 and PTibmmq user accounts to represent users of the client application using the MQSeries transport.

If you have a mainframe to be accessed, install and configure Microsoft Host Integration Server 2004. Use the hissvc local account for HIS LU62 Resync service account. Keep all of the default settings.

In the Computer Management console, add the btssoticomsvc user account to the HIS Runtime Users group that is created when installing Microsoft Host Integration Server 2004.

Install IBM WebSphere MQ for Windows on the local computer. When you install WebSphere MQ for Windows, be sure to do the following:

Select Typical as the Setup Type.

Keep all of the default settings.

Set up the Default Configuration at the end of Prepare WebSphere MQ Wizard. The queue manager will be named as QM\_<your computer name>.

If you are using the IBM WebSphere MQ for Windows platforms v5.3, install the Fix Pack 10 (CSD10).

Insert the Microsoft BizTalk Server 2006 installation disk into the CD-ROM drive. If the BizTalk Server Setup page does not appear, double-click setup.exe located in the root of the installation CD.

On the Microsoft BizTalk Server 2006 Installation Wizard page, click Install Microsoft BizTalk Server 2006 on this computer.

On the Customer Information page, type your user name and organization, and then click Next.

On the License Agreement page, click the Yes, I accept the terms of the license agreement check box to accept the terms of the License Agreement, and then click Next.

On the Component Installation page, select the following component to install:

MQSeries Agent

On the Redistributable Prerequisites page, select Automatically install the redistributable prerequisites from a CAB file check box, click Browse, and then choose the BtsRedistW2k3EN32.cab file which you copied in the previous step

Click Start, point to All Programs, point to Microsoft BizTalk Server 2006, and then click BizTalk MQSeries Agent Configuration Wizard. Perform the following steps:

On the Welcome page, click Next.

On the Application Identity page, select the This User check box, type mqagentsvc in the User box and password, and then click Next.

On the Name of Role page, click Add, and then add btsibmmqadapter local account to the CreatorOwner role. Then, click Next.

On the summary page, click **Next**, and then click **Next** on the completion page.

## How to Install the SO Solution to Production Servers

Follow these steps to prepare the production servers for installing the service-oriented solution on the production servers.

To configure the first BizTalk Server computer

To deploy the MSI to the first BizTalk Server computer

To configure the SOAP receive endpoint on the first BizTalk Server computer

To configure the second BizTalk Server computer

To create the BizTalk Server applications in the Administration Client computer

To import the MSI files for the SO solution in the Administration Client computer

To import binding files and configure BizTalk Server applications in the Administration Client computer

To deploy BAM infrastructure and definition in the Administration Client computer

To create SSO applications and set configuration values to the SSO store in the Administration Client computer

To install the Simple Client application to the Application Client computer

To configure the Web server for SSL on the Legacy System computer

To install the server certificate to Trusted Root Certification Authorities store on the both BizTalk Server computers

To create and configure the MQSeries queues on the Legacy System computer

To install Payment Tracker application to the Legacy System computer

To configure the .NET HIS Integration component to the Legacy System computer

To configure the Stub SAP Web Services to the Legacy System computer

## **Prerequisites**

Before deploying the service oriented solution to the production servers, you must perform the steps in Developer Machine Setup for the SO Solution

### **To configure the first BizTalk Server computer**

Log on to E2E-BTS1 using CONTOSO\btsadmin account.

Configure Microsoft Distributed Transaction Coordinator (MS DTC) for the MQSeries Adapter to access to the remote the MQSeries Agent.

At a command prompt, type the following command, and then press ENTER to open the Component Services console:

```
%SystemRoot%\system32\com\comexp.msc
```

In the Component Services console, expand Component Services, expand Computers, right-click My Computer, and then click Properties.

On the My Computer Properties dialog box, click MSDTC tab, and then click Security Configuration in the MSDTC tab.

On the Security Configuration dialog box, select no authentication required.

Close all of the dialog boxes.

To deploy the MSI to the first BizTalk Server computer

Log on to E2E-BTS1 using CONTOSO\btsadmin account.

Copy the C:\deploy\BizTalkServer folder in your development computer into the C:\ in E2E-BTS1 computer.

Open a command prompt, change the directory to the C:\BizTalkServer folder, and then run the BTSScn.SO.CustomerService.msi file.

On the Select Installation Folder page, type C:\Program Files\SO in the folder box, and then click Next.

On the Welcome page, click Next.

On the Confirmation page, click Next.

On the Information page, click Next.

On the Completion page, click Next.

Install the amqmdnet.dll file to the Global Assembly Cache (GAC):

In Windows Explorer, browse to the <IBM MQSeries Installation Directory>\bin folder, right-click the amqmdnet.dll file, and then click Copy.

In Windows Explorer, go to the %SystemRoot%\assembly folder, and then paste the amqmdnet.dll file to the folder.

To configure the SOAP receive endpoint on the first BizTalk **Server computer**

Log on to E2E-BTS1 using CONTOSO\btsadmin account.

Start Internet Information Services (IIS) Manager as follows: Click Start, point to All Programs, point to Administration Tools, and then click Internet Information Services (IIS) Manager.

In the Internet Information Services (IIS) Manager, right-click Application Pools, select New, and then select Application Pool.

In the Add New Application Pool dialog box, enter an Application pool ID with any value, and then click OK.

In the Internet Information Services (IIS) Manager console, right-click the application pool that you just created, and then click Properties.

On the Properties page, click the Identity tab, select the Configurable check box, type CONTOSO\btssoapisotrusted in the User name box, type the password in the Password box, and then click OK.

In the Internet Information Services (IIS) Manager console, expand Web Sites, expand the Default Web Site, right-click Microsoft.Samples.BizTalk.WoodgroveBank.OrchProxy.Adapter, click Properties, and then set the Application Pool to the application pool which you just created in the previous step

In the Internet Information Services (IIS) Manager, expand Web Sites, expand Default Web Site, right-click Microsoft.Samples.BizTalk.WoodgroveBank.OrchProxy.Inline, click Properties, and then set the Application Pool to the application pool which you just created in the previous step

In Windows Explorer, give permission to the CONTOSO\BizTalk SOAP Isolated Trusted Host Users domain group to modify the %SystemRoot%\Temp folder.

At a command prompt, change the directory to the %SystemDrive%\Inetpub\AdminScripts folder, and then run the following commands to force Internet Information Services to use NTLM for authentication mechanism:

```
cscript                                adsutil.vbs                                set  
W3SVC/1/ROOT/Microsoft.Samples.BizTalk.WoodgroveBank.OrchProxy.Adapter/NTAuthen  
ticationProviders "NTLM"
```

```

cscript                                adsutil.vbs                                set
W3SVC/1/ROOT/Microsoft.Samples.BizTalk.WoodgroveBank.OrchProxy.Inline/NTAuthenti
cationProviders "NTLM"

```

To configure the second BizTalk Server computer

Log on to E2E-BTS2 using CONTOSO\btsadmin account.

Repeat the same procedures as the first BizTalk Server computer for the second BizTalk Server computer.

To create the BizTalk Server applications in the Administration Client computer

Log on to E2E-ADMIN using CONTOSO\btsadmin account.

Open the BizTalk Server Administration console as follows: Click Start, point to All Programs, point to BizTalk Server 2006, and then click BizTalk Server Administration.

In the BizTalk Server Administration console, expand BizTalk Server 2006 Administration, expand BizTalk Group, expand Platform Settings, right-click Hosts, point to New, click Host, and then create the BizTalk Hosts for each of the names in the following table:

Name	Windows group	Options
BizTalkTracking	CONTOSO\BizTalk Tracking Host Users	<b>Allow Host Tracking</b> <b>Make this the default host in the group</b> <b>In-Process</b> <b>Authentication Trusted</b>
SOAdapterVersionOrchestrations	CONTOSO\BizTalk SO Adapter Version Host Users	<b>In-Process</b> <b>Authentication Trusted</b>
SOInlineVersionOrchestrations	CONTOSO\BizTalk SO Inline Version Host Users	<b>In-Process</b> <b>Authentication Trusted</b>
SOAPAdapterTrusted	CONTOSO\BizTalk SOAP Trusted Host Users	<b>In-Process</b> <b>Authentication Trusted</b>
SOAPIsolatedTrusted	CONTOSO\BizTalk SOAP Isolated Trusted Host Users	<b>Isolated</b> <b>Authentication Trusted</b>

MQSeriesAdapter	CONTOSO\BizTalk Adapter Host Users	MQSeries	<b>In-Process</b> <b>Authentication Trusted</b>
-----------------	------------------------------------	----------	--

In the BizTalk Server Administration console, expand BizTalk Server 2006 Administration, expand BizTalk Group, expand Platform Settings, expand Adapters, and then change the send and receive hosts for the adapters listed in the following table:

Adapter	Host name
MQSeries Send Handler	MQSeriesAdapter
MQSeries Receive Handler	MQSeriesAdapter
SOAP Send Handler	SOAPAdapterTrusted

In the **BizTalk Server Administration** console, expand **BizTalk Server 2006 Administration**, expand **BizTalk Group**, expand **Platform Settings**, right-click **Host Instances**, point to **New**, click **Host Instance**, and then create BizTalk Host Instances for each of the host names listed in the following table:

Host Name	Server Name	Logon
BizTalkTracking	E2E-BTS1	CONTOSO\btstracking
	E2E-BTS2	
SOAdapterVersionOrchestrations	E2E-BTS1	CONTOSO\btssoadapter
	E2E-BTS2	
SOInlineVersionOrchestrations	E2E-BTS1	CONTOSO\btssoinline
	E2E-BTS2	
SOAPAdapterTrusted	E2E-BTS1	CONTOSO\btsssoaptrusted
	E2E-BTS2	
SOAPIsolatedTrusted	E2E-BTS1	CONTOSO\btsssoapisotrusted
	E2E-BTS2	
MQSeriesAdapter	E2E-BTS1	CONTOSO\bt sibmmqadapter

	E2E-BTS2	
--	----------	--

## To import the MSI files for the SO solution in the Administration Client computer

Log on to E2E-ADMIN using CONTOSO\btsadmin account.

Copy the C:\deploy\AdminClient folder from your development computer to the C:\ folder in the E2E-ADMIN computer.

At a command prompt, change the directory to the C:\AdminClient folder, type the following command, and then press ENTER to create the BTSScn.SO.CustomerService application:

```
btstask ImportApp -Package:BTSScn.SO.CustomerService.msi -  
  ApplicationName:BTSScn.SO.CustomerService -Server:E2E-SQL -  
  Database:BizTalkMgmtDb
```

To import binding files and configure BizTalk Server applications in the Administration Client computer

Log on to E2E-ADMIN using CONTOSO\btsadmin account.

At a command prompt, change the directory to the C:\AdminClient folder, open the BTSScn.SO.CustomerService.BindingInfo.xml file using Notepad, and then change the contents as follows:

The Host nodes for the orchestrations of the adapter version are defined in ModuleRef. The Name attribute is Microsoft.Samples.BizTalk.WoodgroveBank.Orchestrations.Adapter. For the orchestrations of the adapter version, change the Name and NTGroupName attributes of every Host node as follows:

The Host nodes for the orchestrations of the inline version are defined in the ModuleRef of which the Name attribute is Microsoft.Samples.BizTalk.WoodgroveBank.Orchestrations.Inline. Change the Name and NTGroupName attributes of every Host nodes for the orchestrations of the adapter version as following:

The SendHandler nodes for the SOAP Adapter include a TransportType node. The Name attribute is SOAP. For the SOAP adapter, change the Name and HostTrusted attributes of every SendHandler nodes as follows:

The ReceiveHandler nodes for the SOAP adapter include a TransportType node. The Name attribute is SOAP. For the SOAP adapter, change the Name and HostTrusted attributes of every ReceiveHandler node as follows:

The SendHandler nodes for the MQSeries Adapter include a TransportType node of which the Name attribute is MQSeries. Change the Name and HostTrusted attributes of every SendHandler nodes for the MQSeries Adapter as following:

The ReceiveHandler nodes for the MQSeries Adapter include a TransportType node. The Name attribute is MQSeries. For the SOAP adapter, change the Name and HostTrusted attributes of every ReceiveHandler node as follows:

In the BizTalk Server Administration console, expand BizTalk Server 2006 Administration, expand, BizTalk Group, expand Applications, right-click BTSScn.SO.CustomerService, point to Import, and then click Bindings.

In the Import Bindings dialog box, browse to the C:\AdminClient folder, select the BTSScn.SO.CustomerService.BindingInfo.xml file, and then click Open.

In the BizTalk Server Administration console, expand BTSScn.SO.CustomerService, click Send Ports, right-click StubSAPWebServicePort in the right pane, and then click Properties.

On the General page, click Configure, type the following URL in the Web services URL box in the Transport Properties dialog box.

`http://E2E-  
LEGACY/Microsoft.Samples.BizTalk.WoodgroveBank.StubSAP/StubSAPWS.asmx`

Close all of the dialog boxes.

In the BizTalk Server Administration console, right-click PaymentTrackingSystemRequestPort in the right pane, and then click Properties.

On the General page, click Configure, type the following queue name in the Queue Definition box in the Transport Properties dialog box:

`MQS://E2E-LEGACY/QM_e2e_legacy/LastPaymentsInputQueue`

Close all of the dialog boxes.

In the BizTalk Server Administration console, right-click PendingTransactionSolicitResponsePort in the right pane, and then click Properties.

On the General page, click Configure, type the following URL in the Web services URL box in the Transport Properties dialog box:

`https://E2E-  
LEGACY/Microsoft.Samples.BizTalk.WoodgroveBank.PendingTransactions/PendTransWS.  
asmx`

Close all of the dialog boxes.

In the BizTalk Server Administration console, right-click InlineSOAMQSendPort in the right pane, and then click Properties.

On the General page, click Configure, type the following queue name in the Queue Definition box on the Transport Properties dialog box.

`MQS://E2E-LEGACY/QM_e2e_legacy/InlineSOAOutputQueue`



Close all of the dialog boxes.

In the BizTalk Server Administration console, expand BTSScn.SO.CustomerService, click Receive Locations, right-click PaymentTrackingSystemOutputQueue in the right pane, and then click Properties.

On the General page, click Configure, type the following queue name in the Queue Definition box in the Transport Properties dialog box:

MQS://E2E-LEGACY/QM\_e2e\_legacy/LastPaymentsOutputQueue

Close all of the dialog boxes.

In the BizTalk Server Administration console, right-click AdapterSOAInputQueue in the right pane, and then click Properties.

On the General page, click Configure, type the following queue name in the Queue Definition box on the Transport Properties dialog box.

MQS://E2E-LEGACY/QM\_e2e\_legacy/AdapterSOAInputQueue

Close all of the dialog boxes.

In the BizTalk Server Administration console, right-click InlineSOAInputQueue in the right pane, and then click Properties.

On the General page, click Configure, type the following queue name in the Queue Definition box in the Transport Properties dialog box:

MQS://E2E-LEGACY/QM\_e2e\_legacy/InlineSOAInputQueue

Close all of the dialog boxes.

To deploy BAM infrastructure and definition in the Administration Client computer

Log on to E2E-ADMIN using CONTOSO\btsadmin account.

Create a BAMConfiguration.xml file in the C:\AdminClient folder using Notepad as follows:

Open a command prompt, type the following command, and then press ENTER:

```
set path=%path%;"%ProgramFiles%\Microsoft BizTalk Server 2006\Tracking"
```

At the command prompt, change the directory to the C:\AdminClient folder, type the following command, and then press ENTER. It will create BAM metadata tables and stored procedures.

```
bm setup-databases -ConfigFile:BAMConfiguration.xml
```

At the command prompt, type the following command, and then press ENTER.

```
bm deploy-all -DefinitionFile: ServiceLevelTracking.xml -Server: E2E-SQL
```

To create SSO applications and set configuration values to the SSO store in the Administration Client computer

Log on to E2E-ADMIN using CONTOSO\btsadmin account.

Open a command prompt, type the following command, and then press ENTER to set the PATH environment variable to Enterprise Single Sign-On.

```
set path=%path%;"%ProgramFiles%\Common Files\Enterprise Single Sign-On"
```

At the command prompt, change the directory to the C:\AdminClient folder, type the following command, and then press ENTER to set the SSO server name.

```
ssomanage -serverall E2E-BTS1
```

At the command prompt, open the PendTransAffApp.xml file using Notepad, and then modify the appUserAccount and appAdminAccount node as follows. The PendTransAffApp.xml file represents the PendingTransactions SSO affiliate application.

At the command prompt, open the PendTransUserMap.xml file using Notepad, and then modify the mapping node as follows. The PendTransUserMap.xml file represents user mappings for the PendingTransactions SSO affiliate application.

Using Notepad, edit the PendTransUserMap.xml to add the following mapping node for the users who will use the client application with MQSeries transport:

At the command prompt, open the PmntTrckAffApp.xml file using Notepad, and then modify the appUserAccount and appAdminAccount node as follows. The PmntTrckAffApp.xml file represents the PaymentTracker SSO affiliate application:

At the command prompt, open the PmntTrckUserMap.xml file using Notepad, and then modify the mapping node as follows. The PmntTrckUserMap.xml file represents user mappings for the PaymentTracker SSO affiliate application.

Using Notepad, edit the PmntTrckUserMap.xml file to add the following mapping node for the users who will use the client application with MQSeries transport:

At the command prompt, open the ConfigStoreApp.xml file using Notepad, and then modify the appUserAccount and appAdminAccount node as follows. The ConfigStoreApp.xml file represents a SSO configuration store application, WoodgroveBank.CustomerService, which stores and manages the configuration settings. These settings will be used by the service-oriented solution.

At the command prompt, open the SetConfigValuesInSSO.cmd file using Notepad, and then edit it as follows. The SetConfigValuesInSSO.cmd file sets the values of the configuration parameters in the SSO configuration store application.

Update PendingTransactionsInlineURL, PaymentTrackingInlineQManager, PaymentTrackingInlineMQChannelDefinition, and StubSAPWebServiceURL variables as follows to point the correct endpoints:

Update the path to the BTSScnSSOApplicationConfig.exe to indicate C:\AdminClient\BTSScnSSOApplicationConfig.exe.

Add the -server option to the command to run the BTSScnSSOApplicationConfig.exe with the -set option. The -server option must be after the SSO Identifier Name,ConfigProperties, as follows:

Delete the command to run the BTSScnSSOApplicationConfig with -get option. The -get option doesn't work with -server option.

At the command prompt, type CreateInitialConfigInSSO.cmd, and then press ENTER. It will create the SSO Affiliate Applications and the SSO configuration store application, and create user mappings for the affiliate applications. Then, it will run the SetConfigValuesInSSO.cmd to set the configuration values for the SSO configuration store application.

At the command prompt, run the following commands to set the user credentials for the Payment Tracker affiliate application. These commands will ask you to enter the passwords for the external users, User1, and btsibmmqadapter.

```
ssomanage -setcredentials CONTOSO\User1 WoodgroveBank.PaymentTracker
```

```
ssomanage -setcredentials CONTOSO\btsibmmqadapter WoodgroveBank.PaymentTraker
```

At the command prompt, run the following commands to set the user credentials for the Pending Transaction affiliate application:

```
ssomanage -setcredentials CONTOSO\User1 WoodgroveBank.PendingTransactions
```

```
ssomanage -setcredentials CONTOSO\btsibmmqadapter  
WoodgroveBank.PendingTransactions
```

To install the Simple Client application to the Application Client computer

Log on to E2E-CLIENT using the local Administrator account.

Copy the C:\deploy\ApplicationClient folder from your development computer to the C:\ folder in the E2E-CLIENT computer.

Install the amqmdnet.dll file to the Global Assembly Cache (GAC):

In Windows Explorer, browse to the <IBM MQSeries Installation Directory>\bin folder, right-click the amqmdnet.dll file, and then click **Copy**.

In Windows Explorer, go to the %SystemRoot%\assembly folder, and then paste the amqmdnet.dll file to the folder.

To configure the Web server for SSL on the Legacy System computer

Log on to E2E-LEGACY using the local Administrator account.

Install Certificate Services:

Click Start, point to Control Panel, and then click Add or Remove Programs.

In the Add or Remove Programs dialog box, click Add/Remove Windows Components.

On the Windows Components Wizard page, select the Certificate Services, click Next, and then follow the on-screen instructions to complete the installation.

Create a certificate request:

In the Internet Information Services (IIS) Manager console, expand the Web Sites, right-click the Default Web Site, click Properties, click the Directory Security tab, and then click Server Certificate.

On the Welcome page of the Web Server Certificate Wizard, click Next.

On the Service Certificate page, select Create a new certificate, and then click Next.

On the Delayed or Immediate Request page, click Prepare the request now, but send it later, and then click Next.

On the Name and Security Settings page, keep all the default settings, and then click Next.

On the Organization Information page, type your company's organization and organizational unit names, and then click Next.

On the Your Site's Common Name page, type your computer name in the Common name box, and then click Next.

On the Geographical Information page, type your geographical information, and then click Next.

On the Certificate Request File Name page, type c:\certreq.txt in the File name box, and then click Next.

On the Request File Summary page, click Next, and then click Finish on the Completion page.

Submit the certificate request to the Certification Authority:

In Internet Explorer, visit the following Web site:

<http://localhost/certsrv>

On the Welcome page, click Request a Certificate, and then click Advanced certificate request on the Request a Certificate page.

On the Advanced Certificate Request page, click Submit a certificate request using a base64 encoded PKCS #10 file or a renewal request using a base64 encoded PKCS #7 file.

Copy all the text from the c:\certreq.txt file that you created in the previous step, paste it to the Saved Request box on the Submit a Certificate Request or Renewal Request page, and then click Submit.

Issue a certificate using Certification Authority management tool:

Click Start, point to Administrative Tools, and then click Certification Authority.

In the Certification Authority console, expand your certification authority's name, and then double-click the Pending Request.

In the right pane of the Certification Authority console, right-click the certificate request that you just submitted, point to All Tasks, and then click Issue.

Close the Certification Authority console.

Download the certificate to the Web server:

In Internet Explorer, visit the following Web site:

<http://localhost/certsrv>

On the Welcome page, click View the status of a pending certificate request.

On the View the Status of a Pending Certificate Request page, click the request that you just created.

On the Certificate Issued page, select either of the encoding schemes, and then click Download certificate.

On the Security Warning dialog box, click Save, and then save the certificate as c:\certnew.cer.

Install the certificate to the Web server:

In the Internet Information Services (IIS) Manager console, expand Web Sites, right-click the Default Web Site that you created the certificate for, and then click Properties.

In the Properties dialog box, click the Directory Security tab, and then click Server Certificate.

On the Welcome page, click Next.

On the Pending Certificate Request page, click Process the pending request and install the certificate, and then click Next.

On the Process a Pending Request page, type c:\certnew.cer, and then click Next.

Click Next on the SSL Port page, click Next on the Certificate Summary page, and then click Finish on the Confirmation page.

To install the server certificate to Trusted Root Certification Authorities store on the both BizTalk Server computers

Log on to E2E-BTS1 using CONTOSO\btsadmin.

In Internet Explorer, go to <http://E2E-LEGACY/certsrv>.

On the Welcome page, click Download a CA certificate, certificate chain, or CRL.

On the Download a CA Certificate, Certificate Chain, or CRL page, select the certificate that you created in the CA certificate list box, click Download CA certificate, and then save the certificate as C:\certnew.cer.

To open the MMC console: Click Start, click Run, type mmc.exe in the Open box, and then press ENTER.

On the Console menu, click Add/Remove Snap-in.

In the Add/Remove Snap-in dialog box, click Add, select the Certificates check box on the Add Standalone Snap-in dialog box, and then click Add.

On the Certificates snap-in page, select the Computer Account check box, and then click Next.

On the Select Computer page, select the Local computer check box, and then click Finish.

Click Close on the Add Standalone Snap-in dialog box, and then click OK to close the Add/Remove Snap-in dialog box.

In the MMC console, expand Certificates, expand Trusted Root Certification Authorities, right-click Certificates, point to All Tasks, and then click Import.

On the Welcome page, click Next.

On the File to Import page, click Browse, and then choose the C:\certnew.cer file that you downloaded previously, click Open, and then click Next.

On the Certificate Store page, click Next.

On the Completion page, click Finish.

Repeat the same steps for the E2E-BTS2 computer.

To create and configure the MQSeries queues on the Legacy System computer

Log on to E2E-LEGACY using the local Administrator account.

Configure Microsoft Distributed Transaction Coordinator (MS DTC) for the MQSeries adapter to access to the remote the MQSeries Agent.

Open a command prompt, type the following command, and then press ENTER to open the Component Services console:

```
%SystemRoot%\system32\com\comexp.msc
```

In the Component Services console, expand Component Services, expand Computers, right-click My Computer, and then click Properties.

In the My Computer Properties dialog box, click MSDTC tab, and then click Security Configuration in the MSDTC tab.

In the Security Configuration dialog box, select no authentication required.

Open the WebSphere MQ Explorer, expand Queue Managers, and then expand the queue manager in which you want to create the queues. Typically, a queue manager is named as QM\_<your computer name>.

In the WebSphere MQ Explorer, right-click Queues, point to New, click Local Queue, and then create the following local queues for the adapter version of the solution:

AdapterSOAInputQueue

AdapterSOAOutputQueue

Repeat the above process to create the following local queues for the inline version:

InlineSOAOutputQueue

InlineSOAInputQueue

Repeat the above process to create the following local queues for the Payment Tracker simulator. (The Payment Tracker simulator is used in both the adapter and inline versions.)

LastPaymentsInputQueue

LastPaymentsOutputQueue

Open a command prompt, and run the following commands to grant the required permissions to the mqagentsvc local account under which the MQSeries Agent COM+ application runs:

```
setmqaut -m QM_e2e_legacy -t qmgr -p mqagentsvc +inq +connect +setid
```

```
setmqaut -m QM_e2e_legacy -n AdapterSOAInputQueue -t queue -p mqagentsvc +get  
+browse +put +inq
```

```
setmqaut -m QM_e2e_legacy -n AdapterSOAOutputQueue -t queue -p mqagentsvc +get  
+browse +put +inq
```

```
setmqaut -m QM_e2e_legacy -n InlineSOAInputQueue -t queue -p mqagentsvc +get  
+browse +put +inq
```

```
setmqaut -m QM_e2e_legacy -n InlineSOAOutputQueue -t queue -p mqagentsvc +get  
+browse +put +inq
```

```
setmqaut -m QM_e2e_legacy -n LastPaymentsInputQueue -t queue -p mqagentsvc +get  
+browse +put +inq +setid
```

```
setmqaut -m QM_e2e_legacy -n LastPaymentsOutputQueue -t queue -p mqagentsvc +get  
+browse +put +inq
```

At the command prompt, run the following commands to grant the required permissions to the btssoinline local account. This local account will allow the inline version of orchestrations to access the MQSeries queues.

```
setmqaut -m QM_e2e_legacy -t qmgr -p btssoinline +inq +connect +setid
```

```
setmqaut -m QM_e2e_legacy -n LastPaymentsInputQueue -t queue -p btssoinline +put  
+setid
```

```
setmqaut -m QM_e2e_legacy -n LastPaymentsOutputQueue -t queue -p btssoinline +get
```

At the command prompt, run the following commands to grant the required permissions to User1 local user account. You don't need to perform this step if you want to use the client application with only the SOAP transport.

```
setmqaut -m QM_e2e_legacy -t qmgr -p User1 +inq +connect +setid
```

```
setmqaut -m QM_e2e_legacy -n AdapterSOAInputQueue -t queue -p User1 +put
```

```
setmqaut -m QM_e2e_legacy -n AdapterSOAOutputQueue -t queue -p User1 +get
```

```
setmqaut -m QM_e2e_legacy -n InlineSOAInputQueue -t queue -p User1 +put
```

```
setmqaut -m QM_e2e_legacy -n InlineSOAOutputQueue -t queue -p User1 +get
```

To install Payment Tracker application to the Legacy System computer

Log on to E2E-LEGACY using the local Administrator account.

Copy the C:\deploy\SOLegacyfolder from your development computer to the C:\ folder in the E2E-LEGACY computer.



Install the amqmdnet.dll file to the Global Assembly Cache (GAC) for the BTSScnSOPaymentTracker.exe in the C:\SOLegacy\PaymentTracker folder:

In Windows Explorer, browse to the <IBM MQSeries Installation Directory>\bin folder, right-click the amqmdnet.dll file, and then click Copy.

In Windows Explorer, go to the %SystemRoot%\assembly folder, and then paste the amqmdnet.dll file to the folder.

To configure the .NET HIS Integration component to the Legacy System computer

Log on to E2E-LEGACY using the local Administrator account.

Open a command prompt, type cmd C:\Program Files\SO, and then press ENTER.

At a command prompt, change the directory to the C:\SOLegacy folder, and then copy the PendingTransactions folder to the C:\Program Files\SO folder.

At a command prompt, type the following command, and then press ENTER. This command will change the default script host to CScript.exe:

```
cscript /H:CScript
```

Create and configure the PendingTransactions Web service. The service oriented solution accesses the mainframe through this Web service.

In the Internet Information Services (IIS) Manager, right-click Application Pools, select New, and then select Application Pool.

In the Add New Application Pool dialog box, enter the Application pool ID (any value), and then click OK.

In the Internet Information Services (IIS) Manager, right-click the application pool that you just created, and then select Properties.

On the Properties page, click the Identity tab, select the Configurable check box, type pendingtranappool in the User name box, the password in the Password box, and then click OK.

At a command prompt, type the following command, and then press ENTER to create PendingTransaction Web application for runtime.

```
iisvdir /create "Default Web Site"  
Microsoft.Samples.BizTalk.WoodgroveBank.PendingTransactions "C:\Program  
Files\SO\PendingTransactions"
```

In the Internet Information Services (IIS) Manager console, expand the Web Sites, expand the Default Web Site, right-click Microsoft.Samples.BizTalk.WoodgroveBank.PendingTransactions, click Properties, and

then set the Application Pool to the application pool which you just created in the previous step.

In the Properties dialog box, click the Directory Security tab, and then click Edit to modify Authentication and access control. Select the Basic authentication (password is sent in clear text) check box, and clear the other Authentication access checkboxes. Close all of the dialog boxes.

In Windows Explorer, give the pendingtranapppool local account permission to modify %SystemRoot%\Temp.

Create a COM+ application for the BTSScn SO TI component:

At the command prompt, run %systemroot%\system32\com\comexp.msc.

In the Component Services console, expand Component Services, expand Computers, expand My Computer, right-click COM+ Application, point to New, and then click Application.

On the Welcome page, click Next, and then click Create an empty application on the Install or Create a New Application page.

Type BTSScn SO TI Component in the Enter a name for the new application box, select Server application as Activation type, and then click Next.

In the Account group box of the Set Application Identity page, select This user, and then type btssoticomsvc and the password in the User and Password boxes. The new COM+ application will run under this user account.

On the Add Application Roles page, click Next.

On the Add Users to Roles page, expand CreatorOwner, click Users, and then click Add

In the Select Users or Groups dialog box, select pendingtranapppool, which is a user account that will be used for accessing the mainframe.

On the Completion page, click Finish.

Create a remote environment to access the mainframe:

Click Start, point to All Programs, point to Microsoft Host Integration Server 2004, and then click TI Manager.

In the TI Manager console, click Transaction Integrator (Configuration), expand Windows Initiated Processing, right-click Remote Environments, point to New, and then click Remote Environment.

On the Welcome page, click Next.

On the Configure a New Remote Environment page, type the Remote Application Name, and then click Next. For this procedure, use Mainframe\_TCP for the name.

On the Configure Host Environment and Programming Model page, select CICS for the Target host and ELM Link for the Programming model, and then click Next.

On the Configure Endpoint TCP/IP page, type the IP address for your mainframe in the IP/DNS address box, and then click Edit to add the port number. Your HIS COM will access the transactions through the endpoint address.

On the Completion page, click Finish.

#### Create the TI Component:

In the TI Manager console, click Transaction Integrator (Configuration), click Windows Initiated Processing, and then click Objects. Right-click Objects, click New, and then click Object.

On the Welcome page, click Next.

On the Specify Or Locate An Object page, click Browse, choose SOHISTIUsingCOM.TLB in the C:\Program Files\SO\HISTISComponent folder, and then click Next.

On the Define Environment Characteristics for The COM Object page, select BTSScn SO TI Component for the COM+ application, and then click Next.

On the Define Remote Environment page, select the remote environment you created for the Remote environment, and then click Next.

On the Creation of WIP Objects page, click Next, and then click Finish on the Completion page.

#### Test the connectivity to the mainframe:

In Windows Explorer, browse to the C:\Program Files\SO\HISTISimpleTester folder, and then double-click the Interop.SOHISTIUsingCOM.dll.reg file. This will add registry values for the HISTISimpleTester application to call the TI component through the Runtime Callable Wrapper (RCW).

In Windows Explorer, run BTSScnSOHISTISComponentSimpleTester.exe.

In the HISTISimpleTester application, click Call Mainframe Program - Using COM. It will return five records from the COBOL application running on the mainframe.

#### To configure the Stub SAP Web Services to the Legacy System computer

In the Internet Information Services (IIS) Manager console, right-click Application Pools, select New, and then select Application Pool.

In the Add New Application Pool dialog box, enter the Application pool ID (any value), and then click OK.

In the Internet Information Services (IIS) Manager console, right-click the application pool that you just created, and then click Properties.

On the Properties page, click the Identity tab, select the Configurable check box, type sapappool in the User name box, type the password in the Password box, and then click OK.

In the Internet Information Services (IIS) Manager console, expand Web Sites, right-click Default Web Site, point to New, and then click Virtual Directory to run Virtual Directory Creation Wizard.

Using the Virtual Directory Creation Wizard, create the following virtual directory for the stub SAP Web service:

Alias = Microsoft.Samples.BizTalk.WoodgroveBank.StubSAP

PATH = C:\Program Files\SO\SAP

Access Permissions = Read, Run scripts

In the Internet Information Services (IIS) Manager console, expand Web Sites, expand Default Web Site, right-click Microsoft.Samples.BizTalk.WoodgroveBank.StubSAP, click Properties, and then modify the settings as follows using the Properties dialog box:

On the Virtual directory tab, set the Application Pool to <YourAppPool>, which you just created.

Click the Directory Security tab, click Edit in the Authentication and access control group box, and then select the Enable Anonymous Access check box. Click OK to exit.

In Windows Explorer, give the sapappool local account permissions to modify %SystemRoot%\Temp.

## How to Run the SO Solution on Production Servers

The following steps describe how to run and validate the Service Oriented Solution on the production servers.

To start the service-oriented solution

To send requests using the client application (adapter version)

To send requests using the client application (inline version)

## Prerequisites

How to Install the SO Solution to Production Servers

### To start the service-oriented solution

Log on to E2E-ADMIN as CONTOSO\btsadmin account.

Open the **BizTalk Server Administration** console as follows: Click **Start**, point to **All Programs**, point to **BizTalk Server 2006**, and then click **BizTalk Server Administration**.

In the **BizTalk Server Administration** console, expand **BizTalk Server 2006 Administration**, expand **BizTalk Group**, expand **Applications**, right-click **BTSScn.SO.CustomerService**, click **Start**, and then click **Start** on the **Start Application** dialog box.

### To send requests using the client application (adapter version)

Log on to E2E-LEGACY as the local Administrator account.

Open a command prompt, change the directory to the C:\SOLegacy\PaymentTracker folder, and then run the following command to start the PaymentTracker simulator:

```
BTSScnSOPaymentTracker.exe LastPaymentsInputQueue  
LastPaymentsOutputQueue <Queue Manager Name> 5 [<Channel Definition>]
```

Leave the Payment Tracker simulator running.

Log on to E2E-CLIENT as CONTOSO\User1 account.

Open a command prompt, change the directory to the C:\ApplicationClient folder, and then run the BTSScnSOSimpleClient.exe.

In the BTSScnSOSimpleClient.exe, send a request by SOAP transport using the as follows:

Type any characters in the **RequestType**, **RequestSource**, and **RequestID** text boxes.

Type any 16-digit number in the **Account Number** text box.

Select **SOAP (WS Call)** and **Adapter** in the **Select Transport and Parameters** group box.

Type the following URL in the **URL** text box, for example:

```
http://E2E-  
BTS/Microsoft.Samples.BizTalk.WoodgroveBank.OrchProxy.Adapter/Custom  
ServicePort.asmx
```

Type **ZipCode** in the **Name** text box under **Authentication Elements**, and then type any characters in the **Value** text box.

Type **CustomerName** in the **Name** text box under **Authentication Elements**, and then type any characters in the **Value** text box.

Click **Get my balance**.

The response is displayed in the **Response** text box: **SUCCESS** appears if the request is handled successfully; an error message appears if the request fails.

In the BTSScnSOSimpleClient.exe, send requests by MQSeries transport as follows:

Type any characters in the **RequestType**, **RequestSource**, and **RequestID** text boxes.

Type a 16-digit number in the **Account Number** text box.

Select **MQSeries** in the **Select Transport and Parameters** group box.

Type **QM\_e2e\_legacy** in the **Queue Manager** text box.

Type **AdapterSOAInputQueue** in the **Input Queue** text box.

Type **AdapterSOAOutputQueue** in the **Output Queue** text box.

Type the following channel definition in the **Channel Definition** box.

**S\_e2e\_legacy/TCP/e2e-legacy(1414)**

Type **ZipCode** in the **Name** text box under **Authentication Elements**, and then type any characters in the **Value** text box.

Type **CustomerName** in the **Name** text box under **Authentication Elements**, and then type any characters in the **Value** text box.

Click **Get my balance**.

The response is displayed in the **Response** text box: **SUCCESS** appears if the request is handled successfully; an error message appears if the request fails.

### To send requests using the client application (inline version)

Log on to E2E-LEGACY as the local Administrator account.

Open a command prompt, change the directory to the C:\SOLegacy\PaymentTracker folder, and then run the following command to start the PaymentTracker simulator:

**BTSScnSOPaymentTracker.exe LastPaymentsInputQueue  
LastPaymentsOutputQueue <Queue Manager Name> 5 [<Channel Definition>]**

Leave the Payment Tracker simulator running.

Log on to E2E-CLIENT as CONTOSO\User1 account.

Open a command prompt, change the directory to the C:\ApplicationClient folder, and then run the BTSScnSOSimpleClient.exe.

In the BTSScnSOSimpleClient.exe, send a request by SOAP transport using the as follows:

Type any characters in the **RequestType**, **RequestSource**, and **RequestID** text boxes.

Type any 16-digit number in the **Account Number** text box.

Select **SOAP (WS Call)** and **Inline** in the **Select Transport and Parameters** group box.

Type the following URL in the **URL** text box, for example:

**http://E2E-BTS/Microsoft.Samples.BizTalk.WoodgroveBank.OrchProxy.Inline/CustomerServicePort.asmx**

Type **ZipCode** in the **Name** text box under **Authentication Elements**, and then type any characters in the **Value** text box.

Type **CustomerName** in the **Name** text box under **Authentication Elements**, and then type any characters in the **Value** text box.

Click **Get my balance**.

The response is displayed in the **Response** text box: **SUCCESS** appears if the request is handled successfully; an error message appears if the request fails.

In the BTSScnSOSimpleClient.exe, send requests by MQSeries transport as follows:

Type any characters in the **RequestType**, **RequestSource**, and **RequestID** text boxes.

Type a 16-digit number in the **Account Number** text box.

Select **MQSeries** in the **Select Transport and Parameters** group box.



Type **QM\_e2e\_legacy** in the **Queue Manager** text box.

Type **InlineSOAInputQueue** in the **Input Queue** text box.

Type **InlineSOAOutputQueue** in the **Output Queue** text box.

Type the following channel definition in the **Channel Definition** box.

**S\_e2e\_legacy/TCP/e2e-legacy(1414)**

Type **ZipCode** in the **Name** text box under **Authentication Elements**, and then type any characters in the **Value** text box.

Type **CustomerName** in the **Name** text box under **Authentication Elements**, and then type any characters in the **Value** text box.

Click **Get my balance**.

The response is displayed in the **Response** text box: **SUCCESS** appears if the request is handled successfully; an error message appears if the request fails.

The screenshot shows the 'Service Oriented Scenario - Simple Client' application window. It is divided into several sections:

- Request:** Includes fields for Request Type, Request Source, Request ID (123456), and Account Number (1234567890120002).
- Authentication Elements:** A table with Name and Value columns. The first row has Name: ZipCode and Value: 90052. The second row has Name: CustomerName and Value: jwoodgrove. There are two empty rows below.
- Select Transport and Parameters:**
  - SOAP (WS Call):** Includes a Web Service Details section with a URL field containing 'jodgrove@bank.OrchProxy/Inline/CustomServicePort.asmx' and radio buttons for Adapter, Inline (selected), and Stub.
  - MQSeries:** Includes an MQSeries Transport Parameters section with fields for Queue Manager (QM\_bpc2x\_015), Input Queue (InlineSOAInputQueue), Output Queue (InlineSOAOutputQueue), and Channel Definition (SYSTEM.DEFAULTCOUNN/TCP/bpc2x\_015/channel-name/transport/hostname[port]).
- Response:** A section displaying response data: Credit Limit: 10000, Account Balance: 36032.8, Last Payment: 26644, Pending Trans: 611.2, Cash Advance Limit: 1000, and Time taken: 659.3805 Milliseconds. Below this is a large text area containing the word 'SUCCESS'.

A 'Get my balance' button is located between the Authentication Elements and the Response sections.

## How to Remove a Version of the SO Solution

In the previous steps, you installed the both versions of the SO solution. The inline version might provide better performance, and the adapter version gives better flexibility. In the adapter version, every messages are routed through the publication and subscription (pub/sub) architecture based on the MessageBox database. The pub/sub architecture allows business processes, message

transformations, and transport details to be loosely-coupled and managed independently. You could deal with the change of business or technical requirements by just configuring the adapter version while the inline version requires rewriting assemblies.

It is recommended to use the adapter version than the inline version if the adapter version provides enough performance to your business requirements.

After testing two versions, you might decide to use one of them. The following procedures describe how to remove an unnecessary version.

## Prerequisites

Developer Machine Setup for the SO Solution

### To remove the inline version of the SO Solution

Remove the orchestrations, ports, host, and host instance for the inline version

Log on to E2E-ADMIN as CONTOSO\btsadmin account.

Run BizTalk Server Administration console as follows: Click Start, point to All Programs, point to BizTalk Server 2006, and then click BizTalk Server Administration.

In the BizTalk Server Administration console, expand BizTalk Group, expand Applications, right-click the BTSScn.SO.CustomerService, click Stop. In the Stop Application dialog box, select Full Stop - Terminate Instances, and then click Stop.

In the BizTalk Server Administration console, expand the BTSScn.SO.CustomerService, expand Orchestrations, select all of the orchestrations beginning with Microsoft.Samples.BizTalk.WoodgroveBank.Orchestrations.Inline, right-click them, and then click Unenlist. Select the same orchestrations again, right-click them, and then click Remove.

In the BizTalk Server Administration console, expand Send Ports, right-click the InlineSOAMQSendPort, and then click Unenlist, right click InlineSOAMQSendPort again, and then click Delete.

In the BizTalk Server Administration console, expand Receive Locations, select the InlineSOAWebServicesURL and InlineSOAInputQueue, right-click them, and then click Disable. Expand Receive Ports, select the InlineSOAWebServicePort and InlineSOAMQReceivePort, right-click them, and then click Delete.

In the BizTalk Server Administration console, expand BizTalk Group, expand Platform Settings, and then expand Hosts, right-click SOInlineVersionOrchestrations, and then click Delete.

Remove the MQSeries queues for the inline version.

Log on to E2E-LEGACY as local administrator account.

In the WebSphere MQ Explorer, expand the **Queue Managers**, expand QM\_e2e\_legacy, expand Queues

In the WebSphere MQ Explorer, right-click InlineSOAOutputQueue, and then select **Delete**.

In the WebSphere MQ Explorer, right-click InlineSOAInputQueue and select **Delete**.

Delete the user account and group for the inline version.

Log on to E2E-LEGACY as CONTOSO\btsadmin account.

Click **Start**, point to **All Programs**, point to **Administrative Tools**, click **Computer Management**, and then delete local btssoinline account.

Log on to E2E-DC computer using the Domain Administrator account.

Click **Start**, point to **All Programs**, point to **Administrative Tools**, and then click **Active Directory Users and Computers**, and then delete **CONTOSO\BizTalk SO Inline Version Host Users** domain group and **CONTOSO\btssoinline** domain account from E2E-DC domain controller.

Remove applications and resources for the inline version from the BizTalk Servers.

Log on to E2E-BTS1 as CONTOSO\btsadmin account

Click **Start**, point to **All Programs**, point to **Administration Tools**, click **Internet Information Services (IIS) Manager**, expand the **Web Sites**, expand **Default Web Site**, right-click Microsoft.Samples.BizTalk.WoodgroveBank.OrchProxy.Inline, and then click **Delete**.

In the **Internet Information Services (IIS) Manager**, expand the **Application Pools**, right-click the application pool you created for the previous Web application, click **Delete**, and then click **OK** in the confirmation dialog box.

Click **Start**, point to **Control Panel**, click **Add or Remove Programs**, and then uninstall the IBM WebSphere MQ Client for Windows.

In Windows Explorer, browse to the %SystemRoot%\assembly folder, right-click amqmdnet.dll file, and then click **Delete**.

Repeat the same steps for E2E-BTS2.

## **To remove the adapter version of the SO Solution**

Remove the orchestrations, ports, host, and host instance for the adapter version

Log on to E2E-ADMIN as CONTOSO\btsadmin.

Run **BizTalk Server Administration** console as follows: Click **Start**, point to **All Programs**, point to **BizTalk Server 2006**, and then click **BizTalk Server Administration**.

In the **BizTalk Server Administration** console, expand **BizTalk Group**, expand **Applications**, right-click the BTSScn.SO.CustomerService, click **Stop**. In the **Stop Application** dialog box, select **Full Stop - Terminate Instances**, and then click **Stop**.

In the **BizTalk Server Administration** console, expand the BTSScn.SO.CustomerService, expand **Orchestrations**, select all of the orchestrations beginning with Microsoft.Samples.BizTalk.WoodgroveBank.Orchestrations.Adapter, right-click them, and then click **Unenlist**. Select the same orchestrations again, right-click them, and then click **Remove**.

In the **BizTalk Server Administration** console, expand **Send Ports**, right-click the AdapterSOAMQSendPort, and then click **Unenlist**, right click AdapterSOAMQSendPort again, and then click **Delete**.

In the **BizTalk Server Administration** console, expand **Receive Locations**, select the AdapterSOAWebServicesURL and AdapterSOAInputQueue, right-click them, and then click **Disable**. Expand **Receive Ports**, select the AdapterSOAWebServicePort and AdapterSOAMQReceivePort, right-click them, and then click **Delete**.

In the **BizTalk Server Administration** console, expand **BizTalk Group**, expand **Platform Settings**, and then expand **Hosts**, right-click SOAdapterVersionOrchestrations, and then click **Delete**.

Remove the MQSeries queues for the adapter version.

Log on to E2E-LEGACY as local administrator account.

In the WebSphere MQ Explorer, expand the **Queue Managers**, expand QM\_e2e\_legacy, expand Queues

In the WebSphere MQ Explorer, right-click AdapterSOAOutputQueue, and then select **Delete**.

In the WebSphere MQ Explorer, right-click AdapterSOAInputQueue and select **Delete**.

Delete the user account and group for the adapter version.

Log on to E2E-DC computer using the Domain Administrator account.

Click **Start**, point to **All Programs**, point to **Administrative Tools**, and then click **Active Directory Users and Computers**, and then delete **CONTOSO\BizTalk SO Adapter Version Host Users** domain group and **CONTOSO\btssoadapter** domain account from E2E-DC domain controller.

Remove applications and resources for the adapter version from the BizTalk Servers.

Log on to E2E-BTS1 as CONTOSO\btsadmin account

Click **Start**, point to **All Programs**, point to **Administration Tools**, click **Internet Information Services (IIS) Manager**, expand the **Web Sites**, expand **Default Web Site**, right-click `Microsoft.Samples.BizTalk.WoodgroveBank.OrchProxy.Adapter`, and then click **Delete**.

In the **Internet Information Services (IIS) Manager**, expand the **Application Pools**, right-click the application pool you created for the previous Web application, click **Delete**, and then click **OK** in the confirmation dialog box.

Repeat the same steps for E2E-BTS2.

## Business Process Management Solution

The Business Process Management solution shows how to design a BizTalk application to manage a business process such as service order processing. The solution demonstrates how to construct a process manager and provides guidance about dividing a process into distinct stages. The solution also describes how to construct interruptible orchestrations as well as extensive, sophisticated exception handling.

The sections provide an overview of the solution, detailed explanations of the patterns and design choices, and information about building and running the solution.

### In This Section

- Understanding the Business Process Management Solution
- Developer Machine Setup for the BPM Solution
- Developing a Business Process Management Solution

## Understanding the Business Process Management Solution

The solution described in this section presents one way to implement a business process management application. In an ideal business process manager, the parts of the solution representing the business process—the business rules, communicating with specific backend systems, sending response messages—are separate from the infrastructure supporting the process.

In this solution, a cable service ordering system for Southridge Video, the business process is broken into a series of stages. An order manager, which knows nothing about the business rules and backend systems, directs the operation of the stages. The order manager receives orders from an order broker, which can direct orders to several different order managers.

The solution makes extensive use of Microsoft® BizTalk Server 2006 features and shows, among other things, the use of messages internal to the application for coordinating parts of the application.

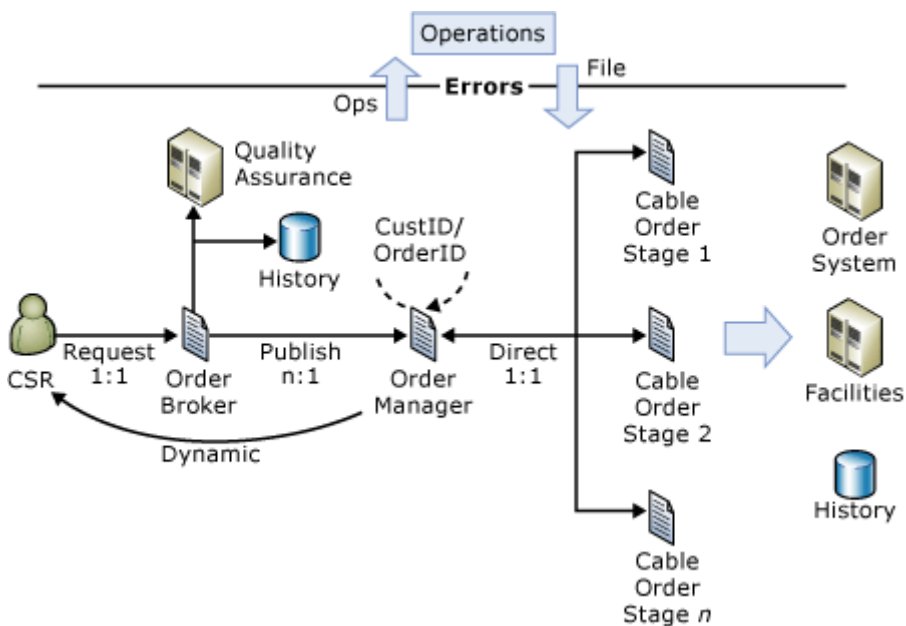
### Reader Guidance

This document assumes that you are familiar with Microsoft BizTalk Server 2006 and Microsoft Visual Studio .NET. It also assumes that you understand basic concepts about enterprise application integration and Web services.

In addition, to read and follow the developer documentation, you should be familiar with how to build applications by using Visual Studio .NET and with performing the following tasks: creating projects, setting references, and debugging and testing BizTalk solutions.

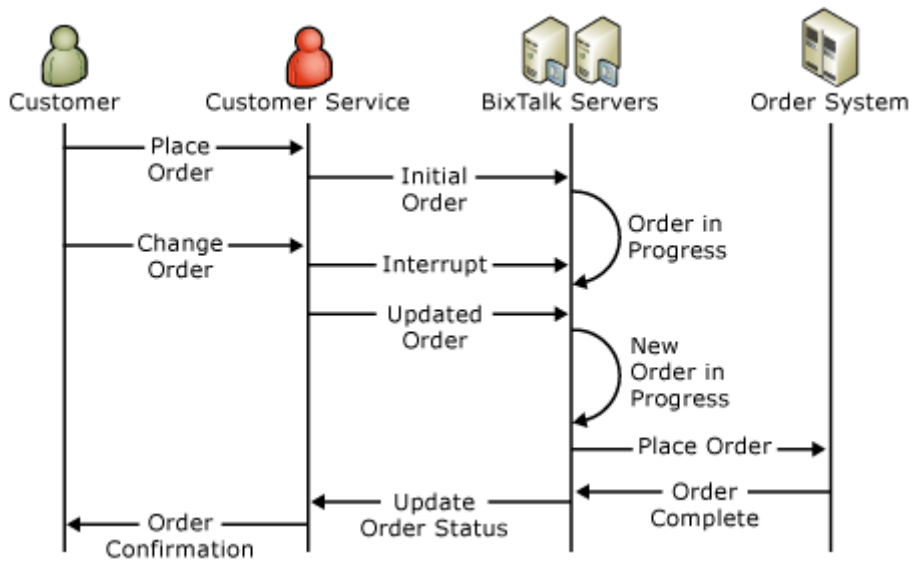
## Ordering Cable Service from Southridge Video

The business process management solution implements a cable service ordering system for Southridge Video. Customers phone into a call center where a customer service representative takes the order and enters it into the order system. The following diagram shows the general flow of an order through the system:



Orders go to the order broker, which sends the order on to the order manager. The order manager runs the processing stages in the proper sequence to process the order. Notice that some kinds of errors go to an operations center for correction and resubmission, and that the solution records the history of each order in a SQL Server table.

The following diagram shows the broad outline of the steps in processing an order.



Notice that an order can be updated as well as cancelled.

### Business Requirements

As with any order system, the ability to update an order is a requirement. However, unlike most retail situations, a cable order may take anywhere from a couple days to almost a year to finish. This affects the solution's design in several ways.

The long lifetime of an order has the potential to produce numerous dehydrated orchestrations. For more information about dehydration, see *Dehydration and Rehydration*. To minimize the storage required, the solution divides the order process into several stages. Thus, if BizTalk Server needs to dehydrate an orchestration, it will be a relatively small orchestration that will take up less storage.

With the long life of orders, there's also the chance that the business process will change while an order is still incomplete. The design of the solution allows for the business to introduce new order processing stages or replace existing stages without disrupting order processing.

The solution also includes an extensive system of interrupts that simplifies order changes or cancellation before they are complete. The solution uses .NET messages to communicate between functional parts of the solution to handle interruptions.

The steps performed by the order broker—validation and publishing the order—are quite generic. The broker in the solution is written so that you can use it with multiple order managers.

Finally, because the system has numerous external dependencies, certain operations can be retried after failure. For example, if a backend system is unavailable and a request to it times out, the solution waits an appropriate interval and retries the request. This portion of the solution makes extensive use of .NET reflection to allow object methods to be retried.

### Business Process Management Solution Resources

Read the following documents for additional information about the business process management solution.

## **Business Process Management Solution Resources**

### **Developing a Business Process Management Solution**

Developers and Software Architects can use this guide to document all code, patterns, architecture, and performance design issues required to build and run the business process management application.

### **Developer Machine Setup for the BPM Solution**

The IT professional with a general understanding of BizTalk Server can use this guide to build and run the Business Process Management application. The guide assumes a general understanding of how the Service Oriented application works in a distributed environment.

## **BizTalk Server 2006 Help**

### **Operations**

BizTalk Server 2006 Operations Guide.

### **Quick Start Guide to Installing and Configuring BizTalk Server 2006**

BizTalk Server 2006 Installation Guide.

### **Security and Protection**

BizTalk Server 2006 Secure Deployment Guide.

### **Troubleshooting**

BizTalk Server 2006 Troubleshooting Guide.

## **Developing a Business Process Management Solution**

The business process management solution is an example of an order system for Southridge Video, a cable service provider. It shows one way to implement the process manager pattern in Microsoft BizTalk Server. The solution uses an orchestration to manage the flow of orders through two satellite orchestrations that implement the business process.

The business requirements of Southridge Video produce a three-part structure: an order broker, a process manager, and the business process itself. Southridge has two separate IT groups involved in the application. A messaging group maintains the corporate messaging infrastructure and provides the components for connecting applications to that infrastructure. Another group writes and maintains applications for specific business processes. Thus, the order broker is separate from the order process manager and process stages so that it can be maintained by a separate group. Because it is a separate component, the order broker can also be used, with some rewriting, to broker orders to multiple process managers. A process manager might be added to support a new business line, such as DSL service.



Southridge Video orders are long running processes: a cable order may take anywhere from a minute to a year to complete. Because an instance of a BizTalk orchestration must run to completion, this means that an orchestration instance could have a lifetime of up to a year.

Southridge needs an architecture for long running processes that allows for application components to change during order processing. Thus, Southridge divides order processing into multiple stages so that an order can complete using the newest process components. For information about how to determine stage boundaries in a business process, see *Building Business Process Management Solutions*.

"Developing a Business Process Management Solution" describes the general workings of the solution, outlines the process management pattern, and describes how the solution implements the pattern. These sections also follow a message through processing in the solution, provides additional implementation details, and tells you how to version and scale the solution.

### **In This Section**

- Patterns in the Business Process Management Solution
- Components of the Business Process Management Solution
- Processing in the Business Process Management Solution
- Implementation Highlights of the Business Process Management Solution
- Building Business Process Management Solutions
- Versioning the Business Process Management Solution
- Business Process Management Solution Reference

## **Deploying the Business Process Management Solution**

The Business Process Management (BPM) solution shows one way to construct a process manager in a BizTalk application. The solution uses a component to select and control the sequence of stages in order processing. The solution takes an order—which may be for a new service, an upgrade, or termination of service—logs it, and acknowledges the order before passing it on for processing. The processing consists of one or more stages that handle the order. Finally, the solution returns a response to the original order request.

This deployment guide describes how to install and run the Business Process Management solution on a single computer.

### **Reader Guidance**

This document assumes that you are familiar with BizTalk Server and Microsoft Visual Studio 2005. It also assumes that you understand basic concepts about enterprise application integration and Web services. Additionally, we recommend that you are familiar with how to build applications by using Visual Studio .NET and that you are familiar with creating projects, setting references, and using the debug mode to debug and test your solution

## In This Section

- Developer Machine Setup for the BPM Solution.
- Deploying the BPM Solution to Production Servers

## Developer Machine Setup for the BPM Solution

The Business Process Management (BPM) solution shows one way to construct a process manager in a BizTalk application. The solution uses a component to select and control the sequence of stages in order processing. The solution takes an order—which may be for new service, an upgrade, or termination of service—logs it, and acknowledges the order before passing it on for processing. The processing consists of one or more stages that handle the order. Finally, the solution returns a response to the original order request.

As a developer, you first get the scenario running on a single computer. After testing functionalities on the single computer, you will discuss how to deploy the solution on production servers with IT professionals. You need to help them to design the system architecture to satisfy service requirements such as high availability, better performance, and easier maintenance. You will then prepare resources for production deployment based on the design.

This deployment guide describes how to install and run the Business Process Management solution on a single computer.

## Reader Guidance

This document assumes that you are familiar with BizTalk Server and Microsoft Visual Studio 2005. It also assumes that you understand basic concepts about enterprise application integration and Web services. Additionally, it is recommended that you are familiar with how to build applications by using Visual Studio and that you are familiar with performing the following tasks: creating projects, setting references, and using the debug mode to debug and test your solution.

## In This Section

- Before Installing the Business Process Management Solution
- How to Install the Business Process Management Solution
- How to Run the Business Process Management Solution

## Before Installing the Business Process Management Solution

The following prerequisites must be installed to deploy the Business Process Management solution on a single computer:

Microsoft Windows Server 2003

Microsoft BizTalk Server 2006

Microsoft Message Queuing (MSMQ) 3.0 (MSMQ HTTP Support)

Microsoft SQL Server 2000 Service Pack 4

Microsoft Internet Information Services 6.0 (WWW and FTP)

ASP.NET 2.0 enabled

Microsoft Visual Studio 2005

Microsoft Excel 2003 with Service Pack 2

## How to Install the Business Process Management Solution

The following steps describe how to prepare the computer for installing the Business Process Management (BPM) solution, and how to install the solution on this computer.

Prepare the computer for installing the Business Process Management Solution

In the preparation step, you create the folders, queues, and SQL database that will be used by the receive and send ports. You also create the two virtual directories for the client application, CSRWebApp, and the OrderBroker proxy Web Service.

Configure the computer for installing the Business Process Management Solution

Install the Business Process Management Solution

### Prepare the computer for installing the Business Process Management Solution

#### To prepare the computer for installing the Business Process Management Solution

Make sure that the **Default Web Site** is configured to use ASP.NET 2.X.

Click **Start**, point to **All Programs**, point to **Administrative Tools**, and then click **Internet Information Services (IIS) Manager**.

In the **Internet Information Services (IIS) Manager**, expand **Web Sites**, right-click, the **Default Web Site**, select **Properties**, and then select **ASP.NET** tab.

Make sure that the **ASP.NET Version** is 2.X.

Click **Start**, point to **All Programs**, point to **Administrative Tools**, and then click **Services**. Using the **Services** console, make sure that the following services are running:

**FTP Publishing**

**Message Queuing**

**World Wide Web Publishing**

Click **Start**, point to **All Programs**, point to **Administrative Tools**, click **Computer Management** console, and then add the BizTalk service account to the local Administrators group.

If you installed Windows SharePoint Services, exclude the (root) of the **Default Web Site** from Windows SharePoint Services Managed Paths as follows: Click **Start**, point to **All Programs**, point to **Administrative Tools**, and then click **SharePoint Central Administration**.

Under **Virtual Server Configuration**, select **Configure virtual server settings**.

On the **Virtual Server List** page, click **Default Web Site**.

On the **Virtual Server Settings** page, click **Define managed paths**.

In the **Included Paths** section of the **Defined Managed Path** page, select **Root** and then click **Remove selected paths**.

At a command prompt, perform an IISReset.

## Configure the computer for installing the Business Process Management Solution

### To configure the computer for installing the Business Process Management Solution

Log off the computer, and then log on to the computer as the BizTalk service account.

Open a command prompt, type the following command, and then press ENTER to set the %BTSSolutionsPath% environment variable to indicate the base folder for the E2E solutions. Then, exit the command prompt.

```
setx      BTSSolutionsPath      "%ProgramFiles%\Microsoft      BizTalk      Server
2006\SDK\Scenarios"
```

Open a command prompt, change the current directory to %BTSSolutionsPath%\BPM\HistoryDB folder, type **CreateDatabase.cmd**, and press ENTER to create the history database.

At a command prompt, run the following command to change the default script host to CScript.exe

```
CScript /H:CScript
```

At a command prompt, run the following command to create the CSRWebApp Web application

```
iisvdir /create "Default Web Site" CSRWebApp
"%BTSSolutionsPath%\BPM\CSRWebApp"
```

At a command prompt, run the following command to create a new IIS virtual directory for OrderBroker\_Proxy.

```
iisvdir /create "Default Web Site" BTSScn.BPM.OrderBroker_Proxy
"%BTSolutionsPath%\BPM\OrderBroker_Proxy"
```

Create a new IIS application pool and set its identity as a user that is a member of the BizTalk Isolated Host Users group and the IIS\_WPG group, as follows:

In the **Internet Information Services (IIS) Manager**, right-click **Application Pools**, select **New**, and then select **Application Pool**.

Type the **Application pool ID** (any value), and then click **OK**.

Right-click the application pool that you created, and then select **Properties**.

Click the **Identity** tab, select **Configurable**, and then enter the user name.

In the **Internet Information Services (IIS) Manager**, expand **Web Sites**, expand, **Default Web Site**, right-click the BTSScn.BPM.OrderBroker\_Proxy, click **Properties**, and then set the **Application Pool** to the application pool which you just created in the previous step.

In the **Internet Information Services (IIS) Manager**, right-click the BTSScn.BPM.OrderBroker\_Proxy, select **Properties**, click **Directory Security** click **Edit** to modify **Authentication and access control**, and then select **Integrated Windows authentication** and make sure all the other check boxes are cleared. Make sure that you disable anonymous access.

In the **Internet Information Services (IIS) Manager**, right-click the CSRWebApp, click **Properties**, and then set the **Application Pool** to the same application pool with the BTSScn.BPM.OrderBroker\_Proxy.

In the **Internet Information Services (IIS) Manager**, right-click the CSRWebApp, select **Properties**, click **Directory Security**, click **Edit** in the **Authentication and access control** section, and then select **Integrated Windows authentication** and make sure all other check boxes are cleared. Make sure that you disable anonymous access.

Reset IIS to make sure all these changes take effect immediately. To do this, run iisreset at a command prompt.

At a command prompt, change the current folder to the %BTSolutionsPath%\BPM\Scripts, type **CreateQueues.vbs**, and then press ENTER to create the following private queues.

Name	Transactional	Transport protocol
ToFacilitiesQ	Yes	Native
FromFacilitiesQ	Yes	Native
FromFixedOrdersQ	Yes	Native
ToServicingSystemQ	Yes	Native

ToCSRSysSystemQ	No	HTTP
ToVendorSystemQ	No	HTTP

At a command prompt, change the current folder to the %BTSSolutionsPath%\BPM\Scripts, type **CreateTestDirectories.cmd**, and then press ENTER.

The following folders are created in %SystemDrive%\BPMTest folder

CSRResponse-DSP

VendorResponse-DSP

OrderErrors-SP

ErrorResponse-RP-TestRL

Facilities-SP

Facilities-RP-TestRL

HistoryInsert-SP

HistoryUpdate-SP

Order-RP-TestRL

ServicingSystem-SP

Vendor-RP-Testily

BizTalkErrors-SP

FromVendor folder is created in the %SystemDrive%\Inetpub\ftproot folder.

## Install the Business Process Management Solution

### To install the Business Process Management Solution

At a command prompt, change the current folder to %BTSSolutionsPath%\BPM, type **SetupBPM.bat**, and then press ENTER. The SetupBPM.bat performs the following tasks:

Creates a unique strong name key (SNK) for signing the assemblies of the BPM Solution.

Extracts the public key token from the SNK.

Updates the binding files with the public token.

Builds the BPM solution, and installs OpsAdapter.

Builds the SSOApplicationConfig in the %BTSSolutionsPath%\Common folder.

Deploy the Southridge Video business rules using the Business Rule Engine Deployment Wizard:

Click **Start**, point to **All Programs**, point to **BizTalk Server 2006**, and then click **Business Rules Engine Deployment Wizard**.

On the **Welcome** page, click **Next**.

On the **Deployment Task** page, select **Import and publish Policy/Vocabulary to database from file**, and then click **Next**.

On the **Policy Store** page, keep all other default settings, and then click **Next**.

On the **Import Rules Engine Policy/Vocabulary file** page, click **Browse**, select the DecodeAndValidateOrderRules.xml file in the %BTSSolutionsPath%\BPM\Rules folder, and then click **Next**.

On the **Ready** page, click **Next**, and then on the **Importing Policy/Vocabulary** page, click **Next**.

On the Completion page, select **Run the wizard again** to open the Wizard again, and then click **Finish**.

On the **Welcome** page, click **Next**.

On the **Deployment Task** page, select **DeployPolicy**, and then click **Next**.

On the **Policy Store** page, keep all other default settings, and then click **Next**.

On the **Deploy Policy** page, select **DecodeAndValidateOrder 1.0** in the **Rule Engine Policy** drop-down list, and then click **Next**.

On the **Ready** page, click **Next**, and then on the **Deploying Policy** page, click **Next**.

On the Completion page, click **Finish**.

If you install the BPM solution to a 64-bit computer, then

Open a 32-bit command prompt as follows: Click **Start**, click **Run**, type **%SYSTEMROOT%\SYSWOW64\CMD.EXE**, and then press ENTER.

At the 32-bit command prompt, change the directory to the %BTSSolutionsPath%\BPM\Scripts folder.

Using Notepad, open the CreateSouthridgeVideoApplication.cmd, and then replace "%CommonProgramFiles%\Enterprise Single Sign-On\ssomanage.exe" with "%SystemDrive%\Program Files\Common Files\Enterprise Single Sign-On\ssomanage.exe".

At the 32-bit command prompt, type type **DeployBPM.cmd**, and then press ENTER.

At a command prompt, change the current folder to %BTSSolutionsPath%\BPM\Scripts, type **DeployBPM.cmd**, and then press ENTER. The DeployBPM.cmd performs the following tasks:

Creates BizTalk Applications for the BPM Solution.

Adds references between the applications.

Imports the binding files.

Deploys the BAM definition files.

Registers the SouthridgeVideo event source.

Creates a Single Sign-On (SSO) affiliated application, and saves configuration values to the SSO application.

Click **Start**, point to **All Programs**, point to **Microsoft BizTalk Server 2006**, and then click **BizTalk Server Administration**.

In the **BizTalk Server Administration** console, expand **BizTalk Group**, expand **Applications**, expand **Receive Locations**, right-click **Vendor-RP-RL**, and then click **Properties**.

On the **Properties** dialog box, click **Configure**, and then enter values as the following table on the **Transport Properties** dialog box:

Property Name	Value
Server	localhost
User Name	<BizTalk service account name>
Password	<BizTalk service account password>

Run the BPM Solution. For more information about running the solution, see How to Run the Business Process Management Solution

## How to Run the Business Process Management Solution

The following steps describe how to run and validate the Business Process Management solution on a single computer.

Start the Business Process Management Solution

Run and validate the Business Process Management Solution



## Start the Business Process Management Solution

### To start the Business Process Management Solution

Click **Start**, point to **All Programs**, point to **Microsoft BizTalk Server 2006**, and then click **BizTalk Server Administration**.

In the **BizTalk Server Administration** console, expand **BizTalk Group**, expand **Platform Settings**, expand **Host Instances**, right-click **BizTalkServerApplication**, and then click **Start**.

In the **BizTalk Server Administration** console, expand **BizTalk Group**, and then expand **Applications**.

Right-click **BTSScn.BPM.MessagingApp**, click **Start**, and then click **Start** on the **Start Application** dialog box.

Right-click **BTSScn.BPM.OrderBrokerApp**, click **Start**, and then click **Start** on the **Start Application** dialog box.

Right-click **BTSScn.BPM.CableOrderApp**, click **Start**, and then click **Start** on the **Start Application** dialog box.

Run the Facilities Simulator as follows: Open a command prompt, change the directory to the %BTSSolutionsPath%\BPM\FacilitiesSimulator\bin\debug folder, type **BTSScnBPMFacilities.exe**, and then press ENTER. This application simulates the facilities processing back-end systems at Southridge Video. Then, keep the FacilitiesSimulator running.

In the FacilitiesSimulator, type the following receive and transmit queues:

Name	Value
Receive Queue	.\private\$\ToFacilitiesQ
Transmit Queue	.\private\$\FromFacilitiesQ

In the FacilitiesSimulator, Click **Start**.

Run the Operation Server as follows: Open a new command prompt, change the current directory to the %BTSSolutionsPath%\BPM\OperationsServer\bin\debug folder, type **BTSScnBPMOperations.exe 8881** at the command prompt, and then press ENTER. Then, keep the Operation Server running. This application displays the message received by the Ops Adapter in error cases.

Run the Cable Provisioning System as follows: Open a new command prompt, change the current directory to the %BTSSolutionsPath%\BPM\CableProvisioningSystemServer\bin\debug folder, type **BTSScnBPMProvisioning.exe 8880**, and then press ENTER. Then, keep the Cable Provisioning System running. This application simulates a beck-end order system, and displays the final orders.

## Deploying the BPM Solution to Production Servers

### In This Section

- Prerequisites of the BPM Solution Production Servers
- Preparing the BPM Solution to be Deployed
- Preparing the Production Servers for the BPM Solution
- Installing the BPM Solution to Production Servers
- Running the BPM Solution on Production Servers

**Prerequisites of the BP**The Business Process Management Solution can consist of different computers, and each computer plays an integral part in the overall sample. This section describes the hardware and software prerequisites for each computer.

#### Domain Controller

Windows Server 2003 Enterprise Edition

#### Two BizTalk Servers

Windows Server 2003 Enterprise Edition

BizTalk Server 2006

Two Network Interface Cards (NIC) per server for Network Load Balancing (NLB)

#### Two SQL Servers for BizTalk Servers

Windows Server 2003 Enterprise Edition

SQL Server 2000 and SP4

Two NICs per server for SQL Cluster

A shared storage resource

#### Administration Client for BizTalk Servers

Windows Server 2003 Enterprise Edition

BizTalk Server 2006

#### Web Server for the BPM Solution

Windows Server 2003 Enterprise Edition

.NET Framework 2.0

External legacy system

Windows Server 2003 Enterprise Edition

.NET Framework 2.0

SQL Server 2000 and SP4

## Preparing the BPM Solution to be Deployed

The following steps describe how to prepare the resources to be deployed to the production servers.

To prepare resources to be deployed to BizTalk Servers

To prepare resources to be used from Administration Client

To prepare resources for BPM Web server

To prepare resource for legacy system

### Prerequisites

Before deploying the BPM Solution to the production servers, you have to finish Developer Machine Setup for the BPM Solution

### To prepare resources to be deployed to BizTalk Servers

Make sure to compile the BPM solution with Release configuration.

Search and terminate all the service instances of the BPM solution.

Stop the host instance running the BPM Solution. For more information on how to stop the host instance, see [How to Stop a Host Instance](#).

Click **Start**, click **Run**, type cmd and press ENTER.

At the command prompt, type the following command, and then press ENTER in order to set %BTSSolutionsPath% environment variable to indicate the base folder of the E2E solutions. Then, exit the command prompt.

```
setx BTSSolutionsPath %ProgramFiles%\Microsoft BizTalk Server 2006\SDK\Scenarios
```

Click Start, click All Programs, click Microsoft Visual Studio 2005, click Visual Studio Tools, and then click Visual Studio 2005 Command Prompt.

At the Visual Studio 2005 Command Prompt, change the directory to %BTSSolutionsPath%\Common\SSOApplicationConfig, and then run the following command line to compile the SSOApplicationConfig.sln.

```
devenv SSOApplicationConfig.sln /Build "Release|AnyCPU" /Out Build.log
```

At the Visual Studio 2005 Command Prompt, change the directory to %BTSSolutionsPath%\BPM\OpsAdapter, and then run the following command line to compile the OpsAdapter.sln.

```
devenv OpsAdapter.sln /Build "Release|AnyCPU" /Out OpsAdapterBuild.log
```

At the Visual Studio 2005 Command Prompt, change the directory to %BTSSolutionsPath%\BPM, and then run the following command line to compile and deploy the BPM Solution.

```
devenv Microsoft.Samples.BizTalk.SouthridgeVideo.sln /Build "Deployment| Mixed Platforms" /Out Build.log
```

```
devenv Microsoft.Samples.BizTalk.SouthridgeVideo.sln /Deploy "Deployment| Mixed Platforms" /Out Build.log
```

```
btstask AddResource -ApplicationName: BTSScn.BPM.MessagingApp -Type: Assembly -  
OverWrite -Options: GacOnAdd,GacOnInstall -  
Source: SchemaClasses\bin\Release\Microsoft.Samples.BizTalk.SouthridgeVideo.Schema  
Classes.dll
```

```
btstask AddResource -ApplicationName: BTSScn.BPM.MessagingApp -Type: Assembly -  
OverWrite -Options: GacOnAdd,GacOnInstall -  
Source: Utilities\bin\Release\Microsoft.Samples.BizTalk.SouthridgeVideo.Utilities.dll
```

```
btstask AddResource -ApplicationName: BTSScn.BPM.MessagingApp -Type: Assembly -  
OverWrite -Options: GacOnAdd,GacOnInstall -  
Source: ServiceLevelTracking\bin\Release\Microsoft.Samples.BizTalk.SouthridgeVideo.S  
erviceLevelTracking.dll
```

```
btstask AddResource -ApplicationName: BTSScn.BPM.MessagingApp -Type: Assembly -  
OverWrite -Options: GacOnAdd,GacOnInstall -  
Source: OperationsClient\bin\Release\Microsoft.Samples.BizTalk.SouthridgeVideo.Operat  
ionsClient.dll
```

```
btstask AddResource -ApplicationName: BTSScn.BPM.MessagingApp -Type: Assembly -  
OverWrite -Options: GacOnAdd,GacOnInstall -  
Source: OperationsHandler\bin\Release\Microsoft.Samples.BizTalk.SouthridgeVideo.Ope  
rationsHandler.dll
```

```
btstask AddResource -ApplicationName: BTSScn.BPM.MessagingApp -Type: Assembly -  
OverWrite -Options: GacOnAdd,GacOnInstall -  
Source: IOperationsSystem\bin\Release\Microsoft.Samples.BizTalk.SouthridgeVideo.IOpe  
rationsSystem.dll
```

```
btstask AddResource -ApplicationName: BTSScn.BPM.CableOrderApp -Type: Assembly -  
OverWrite -Options: GacOnAdd,GacOnInstall -
```

Source: CableProvisioningSystemClient\bin\Release\Microsoft.Samples.BizTalk.SouthridgeVideo.CableProvisioningSystemClient.dll

btstask AddResource -ApplicationName: BTSScn.BPM.CableOrderApp -Type: Assembly - OverWrite -Options: GacOnAdd,GacOnInstall -  
Source: OrderHandler\bin\Release\Microsoft.Samples.BizTalk.SouthridgeVideo.OrderHandler.dll

btstask AddResource -ApplicationName: BTSScn.BPM.CableOrderApp -Type: Assembly - OverWrite -Options: GacOnAdd,GacOnInstall -  
Source: IOrderHandler\bin\Release\Microsoft.Samples.BizTalk.SouthridgeVideo.IOrderHandler.dll

At a command prompt, create BizTalkServer folder under C:\deploy.

At a command prompt, copy %BTSolutionsPath%\BPM\Scripts\CreateTestDirectories.cmd to the c:\deploy\BizTalkServer folder.

Open BizTalk Administration Console.

Create the MSI file for BTSScn.BPM.MessagingApp Application

Navigate to **Console Root->BizTalk Server 2006 Administration->BizTalk Group->Applications**. Right click on BTSScn.BPM.MessagingApp, and select **Export->MSI files....**

Click **Next**, unselect **Bindings** and **System.BizTalk:BAM** on Select Resources page, and click **Next** until Destination page.

Type C:\deploy\BizTalkServer\MessagingApp.msi as MSI file to be generated on Destination pane, and click **Export**.

Create the MSI file for BTSScn.BPM.OrderBrokerApp Application

Export the MSI file for BTSScn.BPM.OrderBroker Application.

Unselect the **Bindings** on Select Resources page.

Type c:\deploy\BizTalkServer\OrderBrokerApp.msi as MSI file to be generated on Destination page.

Create the MSI file for BTSScn.BPM.CableOrderApp Application

Export the MSI file for BTSScn.BPM.CableOrderApp Application.

Unselect the **Bindings** on Select Resources page.

Type c:\deploy\BizTalkServer\CableOrderApp.msi as MSI file to be generated on Destination page.

Prepare the assemblies for OpsAdapter.

Create OpsAdapter folder in the BizTalkServer folder.

Copy the following files in the OpsAdapter\bin folder to the OpsAdapter folder.

Microsoft.Samples.BizTalk.Adapter.Common.dll

Microsoft.Samples.BizTalk.SouthridgeVideo.OpsAdapterMgmt.dll

Microsoft.Samples.BizTalk.SouthridgeVideo.OpsTxAdapter.dll

Microsoft.Samples.BizTalk.SouthridgeVideo.OpsAdater.IOpsAIC.dll

Copy Register\_Ops\_Adapter.vbs from <BizTalk Installation Folder>\Sdk\Scenarios\BPM\OpsAdapterfolder to C:\Deploy\BizTalkServer\OpsAdapter folder.

### To prepare resources to be used from Administration Client

Make sure to compile the BPM solution with Release configuration.

Create AdminClient folder under the C:\deploy.

At a command prompt, copy  
%BTSSolutionsPath%\BPM\Rules\DecodedAndValidateOrderRules.xml to the  
C:\deploy\AdminClient folder.

Create the binding file for BTSScn.BPM.MessagingApp.Test Application

Navigate to **Console Root->BizTalk Server 2006 Administration->BizTalk Group->Applications**. Right click on BTSScn.BPM.MessagingApp, and select **Export->Bindings....**

Select **Export all bindings from the current application**.

Type c:\deploy\AdminClient\BTSScn.BPM.MessagingApp.Test.BindingInfo.xml as **Export to file** name.

Click **OK**.

Create the binding file for BTSScn.BPM.OrderBrokerApp.Test Application

Keep the settings same as BTSScn.BPM.MessagingApp.Test Application.

Type c:\deploy\AdminClient\BTSScn.BPM.OrderBrokerApp.Test.BindingInfo.xml as **Export to file** name.

Create the binding file for BTSScn.BPM.CableOrderApp.Test Application

Keep the settings same as BTSScn.BPM.MessagingApp.Test Application.

Type c:\deploy\AdminClient\BTSScn.BPM.CableOrderApp.Test.BindingInfo.xml as **Export to file** name.

Create the binding file for BTSScn.BPM.MessagingApp Application

Keep the settings same as BTSScn.BPM.MessagingApp.Test Application.

Type c:\deploy\AdminClient\BTSScn.BPM.MessagingApp.BindingInfo.xml as **Export to file** name.

Create the binding file for BTSScn.BPM.OrderBrokerApp Application

Keep the settings same as BTSScn.BPM.MessagingApp.Test Application.

Type c:\deploy\AdminClient\BTSScn.BPM.OrderBrokerApp.BindingInfo.xml as **Export to file** name.

Create the binding file for BTSScn.BPM.CableOrderApp Application

Keep the settings same as BTSScn.BPM.MessagingApp.Test Application.

Type c:\deploy\AdminClient\BTSScn.BPM.CableOrderApp.BindingInfo.xml as **Export to file** name.

Copy all the files in the C:\deploy\BizTalkServer into the C:\deploy\AdminClient folder.

At a command prompt, copy %BTSSolutionsPath%\BPM\BAM\BAMServiceOrder.xml to the C:\deploy\AdminClient folder.

At a command prompt, copy the scripts files and utility application for SSO configuration in to the c:\deploy\AdminClient folder.

Copy the following files in the %BTSSolutionsPath%\BPM\Scripts folder. They will be used for SSO configuration.

SouthridgeVideoSSOConfiguratioin.xml

CreateSouthridgeVideoApplicatioin.cmd

Copy BTSScnSSOApplicationConfig.exe in the %BTSSolutionsPath%\Common\SSOApplicationConfig\bin.

### **To prepare resources for BPM Web server**

Create BPMWEB folder under C:\deploy folder.

At a command prompt, copy %BTSSolutionsPath%\CSRWebApp folder into the c:\deploy\BPMWEB folder.

**To prepare resource for legacy system**

Create BPMLegacy folder under C:\deploy folder.

Copy HistoryDB\SouthridgeVideoHistory.sql to the BPMLegacy folder.

Prepare BPM Facilities application.

At a command prompt, copy BTSScnBPMFacilities.exe in the  
%BTSSolutonsPath%\BPM\FacilitiesSimulator\bin\Release folder.

Prepare BPM Provisioning application.

Create BTSScnProvisioning folder under the BPMLegacy folder.

Copy BTSScnBPMFacilities.exe in CableProvisioningSystemServer\bin\Release.

Copy Microsoft.Samples.BizTalk.SouthridgeVideo.IOrderHandler.dll in  
IOrderHandler\bin\Release.

Copy Microsoft.Samples.BizTalk.SouthridgeVideo.OrderHandler.dll in  
OrderHandler\bin\Release.

Prepare BPM Operations application.

Create BTSScnBPMOperations folder under the BPMLegacy folder.

Copy BTSScnBPMOperations.exe in OperationsServer\bin\Release.

Copy Microsoft.Samples.BizTalk.SouthridgeVideo.IOperationSystem.dll in  
IOperationSystem\bin\Release folder.

Copy Microsoft.Samples.BizTalk.SouthridgeVideo.OperationsHandler.dll in  
OperationsHandler\bin\Release folder.

**Preparing the Production Servers for the BPM Solution**

The following steps describe how to prepare the production servers for installing the BPM solution.

To create a domain controller

To create global domain security groups and user accounts that the BPM Solution will use.

To configure SQL Cluster

To install the first BizTalk Server

To configure the first BizTalk Server



To install the second BizTalk Server

To configure the second BizTalk Server

To setup TCP/IP for configuring Network Load Balancing (NLB) between BizTalk Servers

To install Administration Client

To configure the Administration Client

To install BizTalk web server

To install legacy system

The following table summarizes the server name and the applications to be deployed.

Network name	Applications/Services	Description
E2E-BTS1	BTSScn.BPM.MessagingApp.Test BTSScn.BPM.OrderBrokerApp.Test BTSScn.BPM.CableOrderApp.Test BTSScn.BPM.MessagingApp BTSScn.BPM.OrderBrokerApp BTSScn.BPM.CableOrderApp FROMFIXEDORDERSQ queue FROMFACILITESQ queue Master Secret Service Business Rule Engine	BizTalk Server
E2E-BTS2	BTSScn.BPM.MessagingApp.Test BTSScn.BPM.OrderBrokerApp.Test BTSScn.BPM.CableOrderApp.Test	BizTalk Server

	BTSScn.BPM.MessagingApp BTSScn.BPM.OrderBrokerApp BTSScn.BPM.CableOrderApp Business Rule Engine	
E2E-SQL1	BizTalk databases SSO database Rule engine database	SQL Server
E2E-SQL2	BizTalk databases SSO database Rule engine database	SQL Server
E2E-ADMIN	BizTalk Administration tools	Administration Client
E2E-WEB	CSRWebApp TOCSRSYSTEMQ queue	The server hosting web application for the BPM solution
E2E-LEGACY	SouthridgeVideoHistory database FromVendor FTP service BTSScnBPMFacilities.exe BTSScnBPMOperations.exe BTSScnBPMProvisioning.exe TOFACILITESQ queue TOSERVICINGSYSTEMQ queue TOVENDORSYSTEMQ queue	The server simulating legacy systems
E2E-DC	Domain controller	Domain controller

## Prerequisites

Before deploying the BPM Solution to the production servers, you have to finish Developer Machine Setup for the BPM Solution

### **To create a domain controller**

The BPM solution assumes that this server is named E2E-DC.

Run **dcpromo.exe** to create a new domain controller, new domain, and new forest.

The full DNS name of the new domain is contoso.com

The domain NetBIOS is CONTOSO.

### **To create global domain security groups and user accounts that the BPM Solution will use**

Create the following new organizational unit using dsadd command line.

BTSAccounts

BTSAccounts\btsntsvc

BTSAccounts\Groups

BTSAccounts\Groups\Global

BTSAccounts\Users

Create the following new domain groups under BTSAccounts\Groups\Global organizational unit using dsadd command line.

SSO Administrators

SSO Affiliate Administrators

BizTalk Server Administrators

BizTalk Server Operators

BizTalk Orderbroker Host Users

BizTalk CableOrder Host Users

BizTalk SQL Adapter Host Users

BizTalk Messaging Queuing Adapter Host Users

BizTalk File Adapter Host Users

BizTalk SOAP Adapter Isolated Host Users

BizTalk FTP Adapter Host Users

BizTalk Tracking Host Users

BizTalk SOAP Adapter Host Users

BizTalk OPS Adapter Host Users

SQL Service Group

Create domain users under BTSAccounts\Users organizational unit and add them to the groups you created. You can use dsadd command line.

Domain user name	Domain name description	Members of
ssoadmin	SSO Administrator	SSO Administrators
btsadmin	BizTalk Administrator	SSO Affiliate Administrators BizTalk Server Administrators SSO Administrators
btsinstall	BizTalk Installation	SSO Administrators
ssoslave1	SSO Service	SSO Administrators
ssomaster	SSO Master Secret	SSO Administrators
sqlservice	SQL Service	SQL Service Group
btsruleupdate	Rule Engine Update Service	
btsoperator	BizTalk Server Operators	BizTalk Server Operators

Create domain users under BTSAccounts\btsntsvc organizational unit and add them to the groups you just created. Check User cannot change password checkbox and uncheck Password never expires and User must change password at next logon checkboxes for the following accounts.

Domain	user	Domain name description	Members of
--------	------	-------------------------	------------

name		
btsorderbroker	BizTalk OrderBroker Orchestration	BizTalk OrderBroker Host Users
btscaleorder	BizTalk CableOrder Orchestration	BizTalk CableOrder Host Users
btsqadapter	BizTalk SQL Adapter	BizTalk SQL Adapter Host Users SSO Administrators
btsmqadapter	BizTalk Messaging Queuing Adapter	BizTalk Messaging Queuing Adapter Host Users SSO Administrators
btsfileadapter	BizTalk File Adapter	BizTalk File Adapter Host Users SSO Administrators
btssoapisolated	BizTalk SOAP Adapter Isolated	BizTalk SOAP Adapter Isolated Host Users
btssoapadapter	BizTalk SOAP Adapter	BizTalk SOAP Adapter Host Users
btsftpadapter	BizTalk FTP Adapter	BizTalk FTP Adapter Host Users SSO Administrators
btstracking	BizTalk Tracking	BizTalk Tracking Host Users
btsopsadapter	BizTalk OPS Adapter	BizTalk OPS Adapter Host Users SSO Administrators

### To configure SQL Cluster

Make sure to join all cluster nodes to contoso.com domain.

Install and configure Windows Server 2003 Clustering Services.

Configure MS DTC.

Make sure to enable Network DTC Access.

Install and Configure SQL Cluster.

The BPM solution assumes that the SQL server name is E2E-SQL.

Use CONTOSO\sqlservice domain user account for SQL Server and SQL Server Agent service.

Use Windows Authentication Mode.

Make sure to configure SQL Server in Auto-start service

Add CONTOSO\btsinstall and CONTOSO\btsadmin to the local administrators group.

Install SQL Server 2000 SP4.

### **To install the first BizTalk Server node.**

The BPM solution assumes that the first BizTalk Server name is E2E-BTS1.

Prepare the redistributable prerequisites CAB file.

Make sure to join the system to contoso.com domain.

Add CONTOSO\btsinstall and CONTOSO\btsadmin to the local Administrators group.

Logon using CONTOSO\btsinstall domain account.

Install ASP.NET, IIS and MSMQ.

Make sure to enable Network DTC Access.

Navigate **Start->Control Panel->Add or Remove Programs**, click and **Add/ Remove Windows Components**.

Double click **Application Server** on Windows Component Wizard.

Make sure **Enable network DTC access** checked.

Exit the all the dialog boxes.

Open command line prompt.

Run the following command line to start Component Services Console.

**%SystemRoot%\system32\com\comexp.msc**

Navigate to **Console Root->Component Services->Computers->My Computer**. Then, Right click on **My Computer** and click **Properties**.

Click **MSDTC**, and click **Security Configuratoin....**

Make sure the following option boxes checked.

**Network DTC Access**

**Allow Remote Clients**

**Allow Remote Administrations**

**Allow Inbound**

**Allow Outbound**

**Enable XA Transactions**

Run BizTalk Server 2006 setup program.

In Component Installation dialog box, select the following components to install:

**Server Runtime**

**Administration Tools**

**Enterprise Single Sign-On Administration**

**Enterprise Single Sign-On Master Secret Server**

**BAM Event API**

**Business Rule Components**

In Redistributable Prerequisites dialog box, select "Automatically install the redistributable prerequisites from a CAB file" option, and choose the pre-downloaded CAB file in the step 1.

**To configure the first BizTalk Server node.**

Run Configuration.exe to start Microsoft BizTalk Server 2006 Configuration.

On the **Microsoft BizTalk Server 2006 Configuration page - Start** page,

Select **Custom configuration**.

Type E2E-SQL as **Database server name**.

Click **Configure**.

Click **Enterprise SSO** in the left pane, and in the right pane:

Select **Enable Enterprise Single Sign-On on this computer** checkbox.

Select **Create a new SSO system** option.

Choose CONTOSO\ssomaster as **Enterprise Single Sign-On Service** account.

Choose CONTOSO\SSO Administrators as **SSO Administrator(s)** role.

Choose CONTOSO\SSO Affiliate Administrators as **SSO Affiliate Administrator(s)** role.

Click **Enterprise SSO Secret Backup in the left pane**, and in the right pane:

Type Secret **backup password**, **Confirm password**, **Password reminder**.

Click **Group** in the left pane, and in the right pane:

Select **Enable BizTalk Server Group on this computer** checkbox.

Select **Create a new BizTalk Group** option.

Choose CONTOSO\BizTalk Server Administrators as **BizTalk Administrators Group**.

Choose CONTOSO\BizTalk Server Operators as **BizTalk Operators Group**.

Click **BizTalk Runtime** in the left pane, and in the right pane:

Select **Register the BizTalk Server runtime components**.

Unselect **Create BizTalk In-Process Host and Host Instance?** checkbox.

Unselect **Create BizTalk Isolated Host and Host Instance?** checkbox.

Click **BizTalk Server Business Rules Engine** in the left pane, and in the right pane:

Select **Enable Business Rule Engine on this computer** checkbox.

Choose CONTOSO\btsruleupdate as **Rule Engine Update Service** account.

Click **Apply Configuration** in the tool bar.

Click **Configure** to start configuration.

### **To install the second BizTalk Server node.**

The BPM solution assumes that the second BizTalk Server name is E2E-BTS2.

Prepare the redistributable prerequisites CAB file.

Make sure to join the system to contoso.com domain.

Add CONTOSO\btsinstall and CONTOSO\btsadmin to the local Administrators group.

Logon using CONTOSO\btsinstall domain account.

Install ASP.NET and IIS.



Make sure to enable Network DTC Access as the first BizTalk Server node.

Run BizTalk Server 2006 setup program.

In Component Installation dialog box, select the following components to install:

**Server Runtime**

**BAM Event API**

**Business Rule Components**

In Redistributable Prerequisites dialog box, select "Automatically install the redistributable prerequisites from a CAB file" option, and choose the pre-downloaded CAB file in the step 1.

**To configure the second BizTalk Server node.**

Run Configuration.exe to start Microsoft BizTalk Server 2006 Configuration.

On the **Microsoft BizTalk Server 2006 Configuration page - Start** page,

Select **Custom configuration**.

Type E2E-SQL as **Database server name**.

Click **Configure**.

Click **Enterprise SSO** in the left pane, and in the right pane:

Select **Enable Enterprise Single Sign-On on this computer** checkbox.

Select **Join an existing SSO System** option.

Choose CONTOSO\ssoslave1 as **Enterprise Single Sign-On Service** account.

Click **Group** in the left pane, and in the right pane:

Select **Enable BizTalk Server Group on this computer** checkbox.

Select **Join an existing BizTalk Group** option.

Click **BizTalk Runtime** in the left pane, and in the right pane:

Select **Register the BizTalk Server runtime components**.

Unselect **Create BizTalk In-Process Host and Host Instance?** checkbox.

Unselect **Create BizTalk Isolated Host and Host Instance?** checkbox.

Click **BizTalk Server Business Rules Engine** in the left pane, and in the right pane:

Select **Enable Business Rule Engine on this computer** checkbox.

Choose CONTOSO\btsruleupdate as **Rule Engine Update Service** account.

Click **Apply Configuration** in the tool bar.

Click **Configure** to start configuration.

### **To setup TCP/IP for configuring Network Load Balancing (NLB) between the BizTalk Servers.**

Configure Network Load Balancing and TCP/IP components of the first BizTalk Server.

The BPM solution assumes that the Full Internet name under Cluster Parameters tab is E2E-BTS.contoso.com.

Configure Network Load Balancing and TCP/IP components of the second BizTalk Servers.

Add New Host for E2E-BTS in the E2E-DC.

### **To install the Administration Client**

The BPM solution assumes that the administration client is E2E-ADMIN.

Prepare the redistributable prerequisites CAB file.

Make sure to join the system to contoso.com domain.

Add CONTOSO\btsinstall and CONTOSO\btsadmin to the local Administrators group.

Logon using CONTOSO\btsinstall domain account.

Make sure to enable Network DTC Access and Network COM+ Access.

Navigate **Start->Control Panel->Add or Remove Programs**, click and **Add/ Remove Windows Components**.

Double click **Application Server** on Windows Component Wizard.

Make sure **Enable network DTC access** and **Enable network COM+ Access** checked.

Exit the all the dialog boxes.

Open command line prompt.

Run the following command line to start Component Services Console.

**%SystemRoot%\system32\com\comexp.msc**

Navigate to **Console Root->Component Services->Computers->My Computer**. Then, Right click on **My Computer** and click **Properties**.

Click **MSDTC**, and click **Security Configuratoin....**

Make sure the following check box enabled.

**Network DTC Access**

**Allow Remote Clients**

**Allow Remote Administrations**

**Allow Inbound**

**Allow Outbound**

**Enable XA Transactions**

Run BizTalk Server 2006 setup program.

In Component Installation dialog box, select the following components to install:

**Documentation**

**Administration Tools**

**Enterprise Single Sign-On Administration**

**Business Rule Component**

In Redistributable Prerequisites dialog box, select "Automatically install the redistributable prerequisites from a CAB file" option, and choose the pre-downloaded CAB file in the step 1.

Run SQL Server 2000 setup program.

Check Client tools only on Installation Definition dialog box in installing SQL Server 2000.

Install SQL Server 2000 SP4.

### **To configure Administration Client.**

Run Configuration.exe to start Microsoft BizTalk Server 2006 Configuration.

On the **Microsoft BizTalk Server 2006 Configuration page - Start** page,

Select **Custom configuration**.

Type E2E-SQL as **Database server name**.

Click **Configure**.

Click **Group** in the left pane, and in the right pane:

Select **Enable BizTalk Server Group on this computer** checkbox.

Click **Apply Configuration** in the tool bar.

Click **Configure** to start configuration.

### **To install BPM web server**

The BPM solution assumes that the web server name is E2E-WEB.

Make sure to join the system to contoso.com domain.

Add CONTOSO\btsinstall and CONTOSO\btsadmin to the local Administrators group.

Logon using CONTOSO\btsinstall domain account.

Install ASP.NET and MSMQ (MSMQ HTTP Support).

Install .NET Framework 2.0.

Create "CSRWebApp Users" local group and User1 local account. This account and group will be used to access CSRWebApp web application.

Add User1 to CSRWebApp Users local group.

### **To install legacy system.**

The BPM solution assumes that the legacy system name is E2E-LEGACY.

Install World Wide Web Service, FTP, and MSMQ (MSMQ HTTP Support)

Install .NET Framework 2.0

Install SQL Server 2000 and SP4.

Select Mixed Mode (Windows Authentication and SQL Server Authentication)

Create btsftpadapter, btssqladapter, and btsopsadapter local user accounts. They must have the same name and passwords in the step "To create global domain security groups and user accounts that the BPM Solution will use".

## Installing the BPM Solution to Production Servers

The following steps describe how to install the BPM solution to the production servers prepared through "Preparing the Production Servers for the BPM Solution".

To configure the first BizTalk Server

To deploy the MSI files to the first BizTalk Server

To install and configure the OpsAdapter to the first BizTalk Server

To configure the SOAP Receive endpoint to the first BizTalk Server

To grant write access for creating an event log entry to the first BizTalk Server

To configure the second BizTalk Server

To deploy the MSI files to the second BizTalk Server

To install and configure the OpsAdapter to the second BizTalk Server

To configure the SOAP Receive endpoint to the second BizTalk Server

To grant write access for creating an event log entry to the second BizTalk Server

To install and configure the OpsAdapter to Administration Client

To create BizTalk Server Applications from Administration Client

To import the MSI files from Administration Client

To deploy Business Rule from Administration Client

To import binding files and configure BizTalk Server applications from Administration Client

To deploy BAM infrastructure and definition file from Administration Client

To create SSO application and set configuration values to the SSO store from Administration Client

To configure the BPM web server

To configure the legacy system

### Prerequisites

Before deploying the BPM Solution to the production servers, you have to finish Developer Machine Setup for the BPM Solution

## To configure the first BizTalk Server

Logon to E2E-BTS1 using CONTOSO\btsadmin account.

Create the following MSMQ private transaction queues:

FROMFIXEDORDERSQ

FROMFACILITIESQ

Grant the **Receive Message** permission to the MSMQ private queues above to CONTOSO\BizTalk Messaging Queuing Adapter Host Users

Copy the C:\deploy\BizTalkServer folder in your developer machine into the C:\.

At a command prompt, execute C:\BizTalkServer\CreateTestDirectories.cmd in order to create the following directories.

%SystemDrive%\BPMTest\CSRResponse-DSP

%SystemDrive%\BPMTest\VendorResponse-DSP

%SystemDrive%\BPMTest\OrderErrors-SP

%SystemDrive%\BPMTest\ErrorResponse-RP-TestRL

%SystemDrive%\BPMTest\Facilities-SP

%SystemDrive%\BPMTest\Facilities-RP-TestRL

%SystemDrive%\BPMTest\HistoryInsert-SP

%SystemDrive%\BPMTest\HistoryUpdate-SP

%SystemDrive%\BPMTest\Order-RP-TestRL

%SystemDrive%\BPMTest\ServicingSystem-SP

%SystemDrive%\BPMTest\Vendor-RP-TestRL

%SystemDrive%\BPMTest\BizTalkErrors-SP

%SystemDrive%\Inetpub\ftproot\FromVendor

**Note** If the Windows system is not installed in the C drive, you should replace %SystemDrive% with C:. You have to match the folder names to the address in the binding files that the BPM solution provides.

Grant the Full Control permission to the folders in the BPM folders, which the CreateTestDirectories.cmd created, to CONTOSO\BizTalk File Adapter Host Users group.

Configure MSDTC for SQL Adapter to access to the remote SQL Server.

Make sure to enable Network DTC Access.

Check **no authentication required** check box on Security Configuration dialog box for MSDTC.

### To deploy the MSI to the first BizTalk Server

Run the MessagingApp.msi in the C:\BizTalkServer folder.

Type C:\Program Files\BPM\MessagingApp\ as the folder name on Select Installation Folder dialog box.

Run the OrderBrokerApp.msi in the C:\BizTalkServer folder.

Type C:\Program Files\BPM\OrderBrokerApp\ as the folder name on Select Installation Folder dialog box.

Run the CableOrderApp.msi in the C:\BizTalkServer folder.

Type C:\Program Files\BPM\CableOrderApp\ as the folder name on Select Installation Folder dialog box.

Open a command prompt, change the current folder to the C:\Program Files\BPM\MessagingApp, type the following command, and then press ENTER.

```
%windir%\Microsoft.NET\Framework\v2.0.50727\installutil  
Microsoft.Samples.BizTalk.SouthridgeVideo.Utilities.dll
```

### To install and configure the OpsAdapter to the first BizTalk Server

Create BTSScn.BPM.OpsAdapter folder in the C:\Program Files\BPM.

Copy the following files in the C:\BizTalkServer folder into the BTSScn.BPM.OpsAdapter folder.

Microsoft.Samples.BizTalk.Adapter.Common.dll

Microsoft.Samples.BizTalk.SouthridgeVideo.OpsAdapterMgmt.dll

Microsoft.Samples.BizTalk.SouthridgeVideo.OpsTxAdapter.dll

Microsoft.Samples.BizTalk.SouthridgeVideo.OpsAdater.IOpsAIC.dll

Register\_Ops\_Adapter.vbs

Open Register\_Ops\_Adapter.vbs using a Notepad and insert the following instruction between "GetFullBinPath = FullBinPath" and "End Function".

**GetFullBinPath = "C:\Program Files\BPM\BTSScn.BPM.OpsAdapter\"**

At a command prompt, run Register\_Ops\_Adapter.vbs.

Copy all the files in the BTSScn.BPM.OpsAdapter folder except Register\_Ops\_Adapter.vbs to the %SystemRoot%\Assembly folder using Windows Explorer to register them to GAC.

### **To configure the SOAP Receive endpoint to the first BizTalk Server**

Add CONTOSO\BizTalk SOAP Adapter Isolated Host Users to local IIS\_WPG group.

Create a new IIS application pool and set its identity as CONTOSO\btsssoapisolated.

Set the BTSScn.BPM.OrderBroker\_Proxy application to use the application pool you just created.

Grant the Modify permission of the Windows\Temp to CONTOSO\BizTalk SOAP Adapter Isolated Host Users.

Open a command prompt and change directory to %SystemDrive%\Inetpub\AdminScripts folder. Then, run the following command to force IIS to use NTLM as authentication mechanism.

**cscript adsutil.vbs set  
W3SVC/1/ROOT/BTSScn.BPM.OrderBroker\_Proxy/NTAuthenticationProviders  
"NTLM"**

### **To configure the second BizTalk Server**

Logon to E2E-BTS2 using CONTOSO\btsadmin account.

Copy the C:\deploy\BizTalkServer folder in your developer machine into the C:\.

At a command prompt, execute C:\BizTalkServer\CreateTestDirectories.cmd in order to create the following directories.

%SystemDrive%\BPMTTest\CSRResponse-DSP

%SystemDrive%\BPMTTest\VendorResponse-DSP

%SystemDrive%\BPMTTest\OrderErrors-SP

%SystemDrive%\BPMTTest\ErrorResponse-RP-TestRL

%SystemDrive%\BPMTTest\Facilities-SP

%SystemDrive%\BPMTTest\Facilities-RP-TestRL



%SystemDrive%\BPMTest\HistoryInsert-SP

%SystemDrive%\BPMTest\HistoryUpdate-SP

%SystemDrive%\BPMTest\Order-RP-TestRL

%SystemDrive%\BPMTest\ServicingSystem-SP

%SystemDrive%\BPMTest\Vendor-RP-TestRL

%SystemDrive%\BPMTest\BizTalkErrors-SP

%SystemDrive%\Inetpub\ftproot\FromVendor

Grant the Full Control permission to the folders in the BPM folders, which the CreateTestDirectories.cmd created, to CONTOSO\BizTalk File Adapter Host Users group.

Configure MSDTC for SQL Adapter to access to the remote SQL Server.

Make sure to enable Network DTC Access.

Check **no authentication required** check box on Security Configuration dialog box for MSDTC.

### **To deploy the MSI files to the second BizTalk Server**

Run the MessagingApp.msi in the C:\BizTalkServer folder.

Type C:\Program Files\BPM\MessagingApp\ as the folder name on Select Installation Folder dialog box.

Run the OrderBrokerApp.msi in the C:\BizTalkServer folder.

Type C:\Program Files\BPM\OrderBrokerApp\ as the folder name on Select Installation Folder dialog box.

Run the CableOrderApp.msi in the C:\BizTalkServer folder.

Type C:\Program Files\BPM\CableOrderApp\ as the folder name on Select Installation Folder dialog box.

Open a command prompt, change the current folder to the C:\Program Files\BPM\MessagingApp, type the following command, and then press ENTER.

**%windir%\Microsoft.NET\Framework\v2.0.50727\installutil  
Microsoft.Samples.BizTalk.SouthridgeVideo.Utilities.dll**

## To install and configure the OpsAdapter to the second BizTalk Server

Create BTSScn.BPM.OpsAdapter folder in the C:\Program Files\BPM.

Copy the following files in the C:\BizTalkServer folder into the BTSScn.BPM.OpsAdapter folder.

Microsoft.Samples.BizTalk.Adapter.Common.dll

Microsoft.Samples.BizTalk.SouthridgeVideo.OpsAdapterMgmt.dll

Microsoft.Samples.BizTalk.SouthridgeVideo.OpsTxAdapter.dll

Microsoft.Samples.BizTalk.SouthridgeVideo.OpsAdater.IOpsAIC.dll

Register\_Ops\_Adapter.vbs

Open Register\_Ops\_Adapter.vbs using a Notepad and insert the following instruction between "GetFullBinPath = FullBinPath" and "End Function".

**GetFullBinPath = "C:\Program Files\BPM\BTSScn.BPM.OpsAdapter\"**

At a command prompt, run Register\_Ops\_Adapter.vbs.

Copy all the files in the BTSScn.BPM.OpsAdapter folder except Register\_Ops\_Adapter.vbs to the %SystemRoot%\Assembly folder using Widows Explorer to register them to GAC.

## To configure the SOAP Receive endpoint to the second BizTalk Server

Add CONTOSO\BizTalk SOAP Adapter Isolated Host Users to local IIS\_WPG group.

Create a new IIS application pool and set its identity as CONTOSO\btsssoapisolated.

Set the BTSScn.BPM.OrderBroker\_Proxy application to use the application pool you just created.

Grant the Modify permission of the Windows\Temp to CONTOSO\BizTalk SOAP Adapter Isolated Host Users.

Open a command prompt and change directory to %SystemDrive%\Inetpub\AdminScripts folder. Then, run the following command to force IIS to use NTLM as authentication mechanism.

```
cscript adsutil.vbs set  
W3SVC/1/ROOT/BTSScn.BPM.OrderBroker_Proxy/NTAuthenticationProviders  
"NTLM"
```

## To install and configure the OpsAdapter to the Administration Client

Logon to E2E-ADMIN using CONTOSO\btsadmin account.

Copy the C:\deploy\AdminClient folder in your developer machine into the C:\.

At a command prompt, type **md C:\Program Files\BPM\BTSScn.BPM.OpsAdapter**, and then ENTER.

Copy the following files in the C:\BizTalkServer folder into the BTSScn.BPM.OpsAdapter folder.

Microsoft.Samples.BizTalk.Adapter.Common.dll

Microsoft.Samples.BizTalk.SouthridgeVideo.OpsAdapterMgmt.dll

Microsoft.Samples.BizTalk.SouthridgeVideo.OpsTxAdapter.dll

Microsoft.Samples.BizTalk.SouthridgeVideo.OpsAdater.IOpsAIC.dll

Register\_Ops\_Adapter.vbs

Open Register\_Ops\_Adapter.vbs using a Notepad.

insert the following instruction between "GetFullBinPath = FullBinPath" and "End Function".

**GetFullBinPath = "C:\Program Files\BPM\BTSScn.BPM.OpsAdapter\"**

Comment out the instruction to run CreateAdapter as following.

**'CreateAdapter "OPS", AdapterClsid, "Operations adapter for BPM Scenario"**

At a command prompt, run Register\_Ops\_Adapter.vbs.

Copy all the files in the BTSScn.BPM.OpsAdapter folder except Register\_Ops\_Adapter.vbs to %SystemRoot%\Assembly folder using Widows Explorer to register them to GAC.

### **To create the BizTalk Server Applications from Administration Client**

Logon to E2E-ADMIN using CONTOSO\btsadmin account.

Run **BizTalk Server Administration** console as follows: Click **Start**, point to **All Programs**, point to **BizTalk Server 2006**, and then click **BizTalk Server Administration**.

Click Connect to an existing group on the right pane.

### **To import the MSI files for the BPM Solution from Administration Client**

Open a command prompt, and then change current directory to the C:\AdminClient folder.

At the command prompt, run the following command to add BTSScn.BPM.MessagingApp.Test BizTalk application.

**btstask AddApp -ApplicationName:BTSScn.BPM.MessagingApp.Test -Server:E2E-SQL -Database:BizTalkMgmtDb**

At the command prompt, run the following command to add BTSScn.BPM.OrderBrokerApp.Test BizTalk application.

```
btstask AddApp -ApplicationName:BTSScn.BPM.OrderBrokerApp.Test -Server:E2E-SQL -Database:BizTalkMgmtDb
```

At the command prompt, run the following command to add BTSScn.BPM.CableOrderApp.Test BizTalk application.

```
btstask AddApp -ApplicationName:BTSScn.BPM.CableOrderApp.Test -Server:E2E-SQL -Database:BizTalkMgmtDb
```

At the command prompt, run the following command to import MessagingApp.msi file as following

```
btstask ImportApp -Package:MessagingApp.msi -  
ApplicationName:BTSScn.BPM.MessagingApp -Server:E2E-SQL -  
Database:BizTalkMgmtDb
```

At the command prompt, run the following command to import OrderBrokerApp.msi file as following

```
btstask ImportApp -Package:OrderBrokerApp.msi -  
ApplicationName:BTSScn.BPM.OrderBrokerApp -Server:E2E-SQL -  
Database:BizTalkMgmtDb
```

At the command prompt, run the following command to import CableOrderApp.msi file as following

```
btstask ImportApp -Package:CableOrderApp.msi -  
ApplicatonName:BTSScn.BPM.CableOrderApp -Server:E2E-SQL -  
Database:BizTalkMgmtDb
```

### **To deploy Business Rule from the Administration Client**

Run Business Rules Engine Deployment Wizard.

Make sure to select **Import and publish Policy/Vocabulary to database from file**.

Select E2E-SQL as SQL Server Name and BizTalkRuleEngineDb as BizTalk Management database on selected server.

Select the C:\AdminClient\DecodeAndValidateOrderRules.xml.

Before finishing the wizard, select **Run the wizard again**.

Select the **DeployPolicy** option.

Select E2E-SQL as SQL Server Name and BizTalkRuleEngineDb as BizTalk Management database on selected server.

Select the policy **DecodeAndValidateOrder 1.0** from the drop-down list.

Click **Finish** to close the wizard.

**To import binding files and configure BizTalk Server applications from Administration Client.**

Open the following files in the C:\AdminClient folder using a Notepad.

BTSScn.BPM.MessagingApp.Test.BindingInfo.xml

BTSScn.BPM.OrderBrokerApp.Test.BindingInfo.xml

BTSScn.BPM.CableOrderApp.Test.BindingInfo.xml

BTSScn.BPM.MessagingApp.BindingInfo.xml

BTSScn.BPM.OrderBrokerApp.BindingInfo.xml

BTSScn.BPM.CableOrderApp.Test.BindingInfo.xml

Edit the BTSScn.B2B.Hub.BindingInfo.xml file to change all the ReceiveHandler nodes for the File Adapter as following

Modify every instances of the SendHandler nodes for the File Adapter in the  
BTSScn.BPM.MessagingApp.Test.BindingInfo.xml,  
BTSScn.BPM.OrderBrokerApp.Test.BindingInfo.xml and .  
BTSScn.BPM.CableOrderApp.Test.BindingInfo.xml.

Change BizTalkServerApplication in the Name attribute for File Adapter send handler as following

Modify host names in the BTSScn.BPM.MessagingApp.BindingInfo.xml.

Change the Name attribute of two SendHandler nodes for OpsAdapter as following:

Change the Name attribute of a ReceiveHandler node for the File Adapter as following:

Change the Name attribute of a ReceiveHandler node for the MSMQ Adapter as following:

Modify host names in the BTSScn.BPM.OrderBrokerApp.BindingInfo.xml.

Change the Name and NTGroupName attributes of every Host nodes under the ModuleRefCollection node in the BTSScn.BPM.OrderBrokerApp.BindingInfo.xml as following:

Change the Name attribute of a SendHanlder node for the SQL Adapter as following:

Change the Name attribute of a SendHandler node for the MSMQ Adapter as following:

Change the Name attributes of two ReceiveHandler nodes for the File Adapter as following:

Change the Name attribute of a ReceiveHandler node for the FTP Adapter as following:

Change the Name attribute of a ReceiveHandler node for the SOAP Receive Adapter as following:

Modify host names in the BTSScn.BPM.CableOrderApp.BindingInfo.xml.

Change the Name and NTGroupName attributes of every Host nodes under the ModuleRefCollection node in the BTSScn.BPM.CableOrderApp.BindingInfo.xml as following:

Change the Name attribute of a SendHandler node for the MSMQ Adapter as following:

Change the Name attribute of a SendHanlder node for the SQL Adapter as following:

Change the Name attribute of a ReceiveHandler node for the MSMQ Adapter as following:

Change the Name attribute of a ReceiveHandler node for the File Adapter as following:

Navigate **Console Root->BizTalk Server 2006 Administration->BizTalk Group->Applications**. Then, right click on BTSScn.BPM.MessagingApp.Test, and click **Import->Bindings**. Then, select BTSScn.BPM.MessagingApp.Test.BindingInfo.xml as **File Name**, and click **Open**.

Import BTSScn.BPM.OrderBrokerApp.Test.BindingInfo.xml to BTSScn.BPM.OrderBrokerApp.Test application.

Import BTSScn.BPM.CableOrderApp.Test.BindingInfo.xml to BTSScn.BPM.CableOrderApp.Test application.

Import BTSScn.BPM.MessagingApp.BindingInfo.xml to BTSScn.BPM.MessagingApp application.

Import BTSScn.BPM.OrderBrokerApp.BindingInfo.xml to BTSScn.BPM.OrderBrokerApp application.

Import BTSScn.BPM.CableOrderApp.BindingInfo.xml to BTSScn.BPM.CableOrderApp application.

Configure BizTalkErrors-SP send port of BTSScn.BPM.MessagingApp.

Double click BTSScn.BPM.MessagingApp, click **Send Ports** and double click BizTalkErrors-SP on the right Send Port pane

Click **Configure** on the send port properties dialog box, and type E2E-LEGACY:8881 as **Initialization Data**.

Exit all the dialog boxes.

Configure OrderErrors-SP send port of BTSScn.BPM.MessagingApp.

Double click BTSScn.BPM.MessagingApp, click **Send Ports** and double click OrderErrors-SP on the right Send Port pane

Click **Configure** on the send port properties dialog box, and type E2E-LEGACY:8881 as **Initialization Data**.

Exit all the dialog boxes.

Configure BadOrderResponse-RP-RL receive location of BTSScn.BPM.MessagingApp.

Double click BTSScn.BPM.MessagingApp, click **Receive Locations**, and double click BadOrderResponse-RP-RL on the right Receive Locations pane

Click **Configure** on the receive location properties, and change **Queue** property to FORMATNAME:DIRECT=OS:E2E-BTS1\PRIVATE\$\FROMFIXEDORDERSQ.

Exit all the dialog boxes.

Configure HistoryInsert-SP send port of BTSScn.BPM.OrderBrokerApp.

Double click BTSScn.BPM.OrderBrokerApp, click **Send Ports** and double click HistoryInsert-SP on the right Send Port pane.

Click **Configure** on the send port properties dialog box, and click **Connection String** on **SQL Transport Properties** dialog box.

Input data link properties as following

Property	Value
Select or enter a server name	E2E-LEGACY
Use a specific user name and password	Checked
User name	btssqladapter
Password	The password for btssqladapter
Allow saving password	Checked

Exit all the dialog boxes.

Configure ServicingSystem-SP send port of BTSScn.BPM.OrderBrokerApp.

Double click ServicingSystem-SP on the right Send port pane.

Click **Configure** on the send port properties dialog box, and change **Destination Queue** property to FORMATNAME:DIRECT=OS:E2E-LEGACY\PRIVATE\$\TOSERVICINGSYSTEMQ

Exit all the dialog boxes.

Configure Vendor-RP-RL receive location of BTSScn.BPM.OrderBrokerApp.

Click **Receive Locations** and double click Vendor-RP-RL on the right Receive Locations pane.

Click **Configure** on the receive location properties dialog box.

Input FTP transport properties as following

Property	Value
Server	E2E-LEGACY
User name	btsftpadapter
Password	The password for btsftpadapter

Exit all the dialog boxes.

Configure HistoryUpdate-SP send port of BTSScn.BPM.CableOrderApp.

Double click BTSScn.BPM.CableOrderApp, click **Send Ports** and double click HistoryUpdate-SP on the right Send Port pane.

Click **Configure** on the send port properties dialog box, and click **Connection String** on **SQL Transport Properties** dialog box.

Input data link properties as following

Property	Value
Select or enter a server name	E2E-LEGACY
Use a specific user name and password	Checked
User name	btssqladapter
Password	The password for btssqladapter
Allow saving password	Checked

Exit all the dialog boxes.

Configure Facilities-SP send port of BTSScn.BPM.CableOrderApp.

Double click Facilities-SP on the right Send port pane.



Click **Configure** on the send port properties dialog box, and change **Destination Queue** property to FORMATNAME:DIRECT=OS:E2E-LEGACY\PRIVATE\$\TOFACILITIESQ.

Exit all the dialog boxes.

Configure Facilities-RP-RL receive location of BTSScn.BPM.CableOrderApp.

double click Facilities-RP-RL on the right Receive Locations pane

Click **Configure** on the receive location properties, and change **Queue** property to FORMATNAME:DIRECT=OS:E2E-BTS1\PRIVATE\$\FROMFACILITIESQ.

Exit all the dialog boxes.

### **To deploy BAM infrastructure and definition from Administration Client.**

Create BAMConfiguration.xml in the C:\AdminClient folder as following

Open a command prompt, and then run the following command to set the path.

```
set path=%path%;"%ProgramFiles%\Microsoft BizTalk Server 2006\Tracking"
```

At the command prompt, run the following command to create the databases for BAM infrastructure and DTS packages using the following command

```
bm setup-databases -ConfigFile:BAMConfiguration.xml
```

At the command prompt, run the following command to deploy the BAM definition file.

```
bm deploy-all -DefinitionFile:BAMServiceOrder.xml -Server:E2E-SQL
```

### **To create SSO application and set configuration values to the SSO store from Administration Client.**

At a command prompt, type the following command to set PATH.

```
set path=%path%;"%ProgramFiles%\Common Files\Enterprise Single Sign-On"
```

At the command prompt, change the current folder to the C:\AdminClient, and then run the following command to set the SSO server name.

```
ssomanage -serverall E2E-BTS1
```

Open SouthridgeVideoSSOConfiguration.xml using a Notepad, and then modify it as following:

```
<userGroup>CONTOSO\BizTalk CableOrder Host Users</userGroup>
```

Open CreateSouthridgeVideoApplication.cmd using a Notepad, and then modify it as following:

Change the value of CableProvisioningSystemLocation variable into E2E-LEGACY:8880.

Update the instruction running BTSScnSSOApplicationConfig.exe with the set option as following:

```
C:\AdminClient\BTSScnSSOApplicationConfig -set %ConfigApp% %identity% -  
server E2E-BTS1 CableProvisioningSystemLocation  
%CableProvisioningSystemLocation% TotalStages %TotalStages%  
CacheRefressInterval %CacheRefressInterval%
```

Comment out the following instruction.

```
..\..\common\SSOApplicationConfig\bin\BTSScnSSOApplicationConfig -get  
%ConfigApp% %identity% CableProvisioningSystemLocation TotalStages  
CacheRefreshInterval
```

At the command prompt, run CreateSouthridgeVideoApplication.cmd.

### To configure the BPM web server.

Logon to E2E-WEB using CONTOSO\btsadmin account.

Copy the C:\deploy\BPMWEB folder in your developer machine into the C:\.

Configure MSMQ for BizTalk MSMQ Adapter.

Create TOCSRSysmq MSMQ private non-transactional queue.

Configure CSRWebApp.

Copy the CSRWebApp folder in the C:\BPMWEB folder to the  
%SystemDrive%\Inetpub\wwwroot.

Add CONTOSO\BizTalk SOAP Adapter Isolated Host Users to IIS\_WPG local group.

Grant the Modify permission of the Windows\Temp to CONTOSO\BizTalk SOAP Adapter Isolated Host Users.

Run the IIS Manager and create a new IIS application pool and set its identity as CONTOSO\btssoapisolated.

Select the Default Web Site, right-click the CSRWebApp folder, select properties, and click **Create** to create an application.

Set the CSRWebApp application to use the application pool you just created.

Slick **ASP.NET** and change the ASP.NET version to 2.0.XXX.

Select **Directory Security** and click **Edit** to modify **Authentication and access control**. Check **Integrated Windows authentication** and uncheck everything else. Make sure that you disable anonymous access.

Exit all the dialog boxes.

Open Web.Config in %SystemDrive%\Inetpub\wwwroot\CSRWeb and modify the server name of OrderBrokeOrch\_Proxy as following.

Open CSRMainForm.aspx.cs in Inetpub\wwwroot\CSRWeb and modify the endpoint address for reply as following:

```
orderRequest.Header.SourceSystem = "MSMQ://FORMATNAME:DIRECT=HTTP://E2E-  
WEB/msmq/private$/ToCSRSystemQ;
```

### To configure the legacy system

Login to E2E-LEGACY using local administrator user account.

Copy the C:\deploy\BPMLegacy folder in your developer machine into the C:\.

At a command prompt, type **md C:\Program Files\BPM**, and then ENTER.

Create MSMQ queues for BizTalk MSMQ Adapter.

Create the following MSMQ private transactional queues:

TOFACILITIESQ

TOSERVICINGSYSTEMQ

Create TOVendorSystemQ MSMQ private non-transactional queue.

Create History database for logging.

Open a SQL Query Analyzer, and then run SouthridgeVideoHistory.sql in the C:\BPMLegacy folder.

Exit the SQL Query Analyzer.

Create user account for SQL Adapter and configure security.

Open SQL Server Enterprise Manager.

Navigate to **Console Root->Microsoft SQL Servers->SQL Server Group->(local)(Windows NT)->Security**, and right click **Logins**. Then, select **New Logins**.

Input the following values on SQL Serve Login Properties dialog box.

Property	Value
Name	btssqladapter
SQL Server Authentication	Checked
Database	SouthridgeVideoHistory
Password	This password must match the password used by the SQL Adapter.

Select **Database Access** tab and check **permit** check box for SouthridgeVideoHistory database.

Exit all the dialog boxes.

Navigate to **Console Root->Microsoft SQL Servers->SQL Server Group->(local)(Windows NT)->Database**, and expand SouthridgeVideoHistory node.

Click **Stored Procedure**, and double click UpdateStatus on the right pane.

Select **Permission** button and check **EXEC** permission of btssqladapter.

Exit all the dialog boxes.

Click **Tables** under SouthridgeVideoHistory database, and double click OrderLog table on the right pane.

Select **Permission** button and check **INSERT** and **UPDATE** permission of btssqladapter.

Exit all the dialog boxes.

Install Configure the BPM Provisioning, BPM Operations, and BPM Facilities application.

Create BTSScBPMFacilities folder in the C:\Program Files\BPM, and then copy the BTSScBPMFacilities.exe in the C:\BPMLegacy folder to the BPM folder.

Create BTSScBPMOperations folder in the C:\Program Files\BPM, and then copy the following files in the C:\BPMLegacy folder to the BPM folder.

BTSScBPMOperations.exe

Microsoft.Samples.BizTalk.SouthridgeVideo.IOperationSystem.dll

Microsoft.Samples.BizTalk.SouthridgeVideo.OperationsHandler.dll

Create BTSScBPMProvisioning folder in the C:\Program Files\BPM, and then copy the following files in the C:\BPMLegacy folder to the BPM folder.

BTSScBPMProvisioning.exe

Microsoft.Samples.BizTalk.SouthridgeVideo.IOrderHandler.dll

Microsoft.Samples.BizTalk.SouthridgeVideo.OrderHandler.dll to the folder

Configure external FTP server.

Create a FromVendor folder in the %SystemDrive%\Inetpub\ftproot folder.

Grant btsftpadapter local user account the read and write permission to the %SystemDrive%\Inetpub\ftproot\FromVender folder.

Configure MSDTC for SQL Adapter to access to the local SQL Server.

Make sure to enable Network DTC Access.

Check **no authentication required check** box on Security Configuration dialog box for MSDTC.

Check **Allow Remote Clients** check box on Security Configuration dialog box for MSDTC.

## Running the BPM Solution on Production Servers

The following steps describe how to run and validate the Business Process Management solution on the production servers.

To start the BPM Solution

To run the legacy applications

To run and validate the BPM Solution

### To start the BPM Solution

Logon to E2E-ADMIN as CONTOSO\btsadmin.

Open BizTalk Server Administration Console.

Start BTSScn.BPM.MessagingApp application.

Navigate to **Console Root->BizTalk Server 2006 Administration->BizTalk Group**, right click on BTSScn.BPM.MessagingApp, and select **Start**.

Uncheck BTSScn.BPM.MessagingApp.Test in Referenced applications pane, and, click **Start**.

Start BTSScn.BPM.OrderBrokerApp application.

Right click on BTSScn.BPM.OrderBrokerApp, and select **Start**.

Uncheck BTSScn.BPM.MessagingApp.Test, BTSScn.BPM.OrderBrokerApp.Test in Referenced applications pane, and click **Start**.

Start BTSScn.BPM.CableOrderApp application.

Right click on BTSScn.BPM.CableOrderApp, and select **Start**.

Uncheck BTSScn.BPM.MessagingApp.Test, BTSScn.BPM. CableOrderApp.Test in Referenced applications pane, and click **Start**.

### To run the legacy applications

Logon to E2E-LEGACY as local administrator.

Run the BTSScnBPMFacilities.exe application from C:\Program Files\BPM\BTSScnBPMFacilities folder. This application simulates the facilities processing back-end systems at Southridge Video.

Use the following receive and transmit queues:

Receive Queue -- .\private\$\ToFacilitiesQ

Transmit Queue -- E2E-BTS1\private\$\FromFacilitiesQ

Click **Start** button to run.

Run the Operations Server console application by entering the following at a command prompt from C:\Program Files\BPM\BTSScnBPMOperations folder: **BTSScnBPMOperations.exe 8881**

Run the Cable Provisioning System Server console application by entering the following at a command prompt from the C:\Program Files\BPM\BTSScnBPMProvisioning folder:

**BTSScnBPMProvisioning.exe 8880**

### To run and validate the BPM solution

Logon to E2E-LEGACY as local administrator.

Run the Customer Service Web application by connecting to <http://E2E-WEB/CSRWebApp/CSRMainForm.aspx>

Enter User1 and the password in the logon dialog box.

Enter the Customer ID, Order ID, and Sequence Number, and choose a Service Type Code, then click **Submit Order**.

After a few seconds of processing time, you should see activity on the Cable Provisioning System Server console application.

There are several sample messages in the BPM\SampleMessages folder that can be used to exercise the BPM solution.

## **Business-to-Business Solution**

### **Developing a Business-to-Business Solution**

The business-to-business solution demonstrates a distributor network run by Litware. Brokers or customers can place orders with vendors or manufacturers using the distributor network. The distributor network is an example of the message broker pattern implemented in Microsoft BizTalk Server 2006. The solution uses the Trading Partner Management system to control the routing of messages among brokers and vendors.

This guide describes the general workings of the solution and outlines the message broker pattern and how the solution implements it. The guide also follows a message through processing in the solution, provides additional implementation details, and tells you how to version and scale the solution. The message broker pattern is useful in any context where you have multiple message sources and multiple possible recipients.

#### **In This Section**

- Patterns in the Business-to-Business Solution
- Components of the Business-to-Business Solution
- Processing in the Business-to-Business Solution
- Implementation Highlights of the Business-to-Business Solution
- Business-to-Business Solution Reference

### **Patterns in the Business-to-Business Solution**

This section describes the basic programming patterns making up the Business-to-Business solution. It begins by considering the enterprise patterns involved in the solution, and then develops those patterns into configurations of BizTalk components. The final section catalogs the patterns in the solution whether they are general enterprise patterns, BizTalk Server-specific patterns, or general programming patterns.

#### **In This Section**

- Designing with Patterns: the Business-to-Business Solution
- Translating the Patterns of the Business-to-Business Solution
- Pattern Catalog for the Business-to-Business Solution

### **Designing with Patterns: the Business-to-Business Solution**

The business-to-business solution shows one way to handle business connections in a BizTalk application. The solution uses the Trading Partner Management system in BizTalk Server to

manage message addresses and certificates. The solution takes an order or an order update from a broker and routes it to the appropriate vendor.

### **What Is the Solution's Main Pattern?**

The primary function of the business-to-business solution is to route orders from customers to appropriate vendors and suppliers. The solution is cleanest if the destination of an order and its origin are decoupled; that is, if the customer can submit an order and the solution will route it to the correct vendor or supplier. In addition, if the solution can centralize control over the routing of orders, it is easier to maintain because there is a single place to update customer, vendor, and supplier addresses.

The Message Broker pattern is designed to accomplish exactly this: to keep the destination of a message separate from its origin while maintaining centralized control over the routing of the messages. In this implementation, the solution serves as a hub that receives messages and routes them to appropriate recipients. The solution can also accept queries about order status as well as provide information about individual brokers and vendors.

## **Translating the Patterns of the Business-to-Business Solution**

This section describes how the solution translates the Message Broker pattern into BizTalk Server artifacts.

### **Implementing the Message Broker Pattern**

The Message Broker pattern is not a single component. Rather, it is a collection of components organized so that the sender of an order does not need to know the address of the recipient. This requires some sort of routing mechanism. In the business-to-business solution, the recipient is determined by the content of the order. The solution uses the Trading Partner Management (TPM) system of BizTalk Server to determine the recipient of a message.

When the solution receives a message, it looks up the appropriate profile and sends the message to the appropriate recipient. Based on this routing mechanism, the solution also includes facilities for communicating with the back-end application and the various brokers and vendors.

For more information about the Trading Partner Management system, see *Developing Business Activity Services Solutions* .

## **Pattern Catalog for the Business-to-Business Solution**

The patterns in the business-to-business solution include patterns directly connected to BizTalk Server programming practices, as well as the enterprise integration patterns in preceding sections. The list in this section includes both kinds of patterns.

### **Patterns Types**

The following entries briefly describe each pattern and point to other topics that describe how the solution uses the pattern. In the case of general pattern types, such as a filter, entries point to more general topics.



## Filter Pattern

The filter pattern selects messages that meet particular criteria for processing. In BizTalk Server, the filter pattern almost always becomes a filter expression on a port. For more information about filters on ports, see [Using Filters to Receive Messages](#) .

## Identification by Location Pattern

Using signed messages for orders is the best practice. And the simplest way of using the Trading Partner Management system presupposes signed messages. In the absence of signatures, however, you can retrieve the appropriate identifiers based on the receive location of a message. The solution does this for unsigned messages, such as for EDI messages, using helper methods. For an example, see [Using EDI in the Business-to-Business Solution](#).

## Message Broker Pattern

The Message Broker pattern enables the solution to determine the destination of a message and yet maintain control over the message flow. The business-to-business solution implements a version of the Message Broker pattern by sending orders to the appropriate vendor. For more information about this pattern in the solution, see [Processing Orders in the Business-to-Business Solution](#). For a different approach to message brokering in the Business Process Management solution, see [Processing in the OrderBroker Orchestration](#) .

## Role Links Pattern

The solution makes extensive use of role links to determine message destinations. For general information about role links, see [Using Role Links](#) . For an example of their use in the solution, see "Confirming Orders" in [Processing Orders in the Business-to-Business Solution](#).

## Splitter Pattern

The splitter pattern takes a single message and splits it into multiple messages. Often you combine the splitter pattern with the Aggregator pattern to assemble the results from the split message. In the **ConfirmOrderWithVendors** orchestration the business-to-business solution splits a purchase order into separate orders for individual vendors. For more information about how the solution uses the Splitter pattern, see "Confirming Orders" in [Processing Orders in the Business-to-Business Solution](#).

## Suspend with Retry Pattern

Many of the orchestrations in the solution suspend a message when there is an error. The suspension occurs within a loop so that the orchestration suspends, asks for operator intervention, and then retries the operation a fixed number of times. For an example, see the **GetPurchaseOrder** orchestration described in [Retrieving Orders in the Business-to-Business Solution](#).

## Translator Pattern

The enterprise pattern of a translator—conversion of a message from one form to another form—most often becomes into a BizTalk map. For general information about BizTalk maps, see *Creating Maps Using BizTalk Mapper* .

## Components of the Business-to-Business Solution

This section describes the major BizTalk Server components of the business-to-business solution.

For information about the source files, see *File Inventory for the Business-to-Business Solution*


### Orchestrations in the Business-to-Business Solution

The Business-to-Business solution presents itself as a set of operations or functions that brokers and vendors can use by sending appropriate messages. The **NewOrder** orchestration accepts new orders, verifies the broker's profile by calling the **GetUserProfile** orchestration, uses an HTTP request-response port to send the order to the back-end process, and calls the **ConfirmOrderWithVendors** and **ConfirmOrderWithBroker** orchestrations to verify the order.

The **UpdateOrder** orchestration receives order update messages, calls the **CheckOrderExists** orchestration to verify that there is an order to update, uses an HTTP request-response port to send the update to the back-end process, calls the **GetUserProfile** orchestration to verify the broker's profile, and calls the **ConfirmOrderWithVendors** and **ConfirmOrderWithBroker** orchestrations.

The **EDI XmlToBasket** orchestration provides an EDI front-end to the solution. The orchestration receives an EDI order, converts it, and sends it to the **NewOrder** orchestration.

The **GetOrder** and **GetProfile** orchestrations enable subscribers to the appropriate roles to retrieve order and profile information. The **GetOrder** orchestration uses the **GetPurchaseOrder** orchestration to retrieve order information. Similarly, the **GetProfile** orchestration uses the **GetUserProfile** orchestration to retrieve profile information.

The **UpdateUserProfileBAS** orchestration enables users enlisted in the appropriate roles to update profile information. The orchestration uses the Web service interface to the Trading Partner Management (TPM) database.  Back-end Application

The intended back-end application for the business-to-business solution is Microsoft Commerce Server 2006. However, the solution uses a set of orchestrations in the **OrderSystem** directory to simulate Commerce Server.

The solution communicates with the back-end orchestrations through a single request response port using HTTP.

### Pipelines for the Business-to-Business Solution

All but two of the pipelines in the solution are default pipelines. The two exceptions are custom pipelines created using the Pipeline editor. The **UpdateOrder**, **UpdateUserBASRP**, **GetOrder**,

and **GetProfile** use the **PartnerReceive** pipeline that includes the Party Resolution and MIME/SMIME components. For more information about the Party Resolution component, see [Party Resolution Pipeline Component](#) . For more information about the MIME/SMIME Decoder component, see [MIME/SMIME Decoder Pipeline Component](#) .

The **NewOrder** orchestration uses the **UpdateOrderToNamePR** pipeline on one of its receive locations to handle order messages from the **EDIXmlToBasket** orchestration. The pipeline includes the XML Disassembler component, as well as a custom component, the **CustomPartyResolution** component. The **CustomPartyResolution** component looks up party information by organization or alias.

## Trading Partner Management in the Business-to-Business Solution

The solution includes several Microsoft InfoPath forms to help in managing trading partners. For more information about Trader Partner Management, see [Partner Management in BizTalk Explorer](#) and [Tutorial 4: Trading Partner Management](#).

### Other Assemblies

The **Utilities** assembly includes objects and methods used throughout the solution. The objects and methods are used to convert XML to a string and back, and support the use of Trading Partner Management, and Enterprise Single Sign-On (SSO) for configuration values.

## Processing in the Business-to-Business Solution

You can best understand how the Business-to-Business Solution works by thinking of it as a collection of processes. These processes include:

Submitting orders

Inquiring about orders

Updating orders

Inquiring about trading partners

Updating trading partner information.

Submitting orders using EDI messages

The different processes often share components. For example, EDI orders come in to an orchestration that converts them to Commerce Server baskets and then submits the orders to a common order processing component. This section describes the workflow through each of the major processes.

### In This Section

- Processing Orders in the Business-to-Business Solution

- Getting Profile Information in the Business-to-Business Solution
- Retrieving Orders in the Business-to-Business Solution
- Updating Orders in the Business-to-Business Solution
- Retrieving Partner Profiles in the Business-to-Business Solution
- Updating Partner Profiles in the Business-to-Business Solution
- Using EDI in the Business-to-Business Solution

## Processing Orders in the Business-to-Business Solution

This section describes how the solution processes all orders except for EDI orders. For information about how the solution processes EDI orders, see Using EDI in the Business-to-Business Solution.

### Processing New Orders

The basic function of the **NewOrder** orchestration is to prepare and send the order to the back-end processing system, and to send confirming e-mail messages to the broker and vendors.

New orders arrive in a file folder (c:\Litware\BrokerNew) as XML files using the Commerce Server basket schema (Basket.xsd). The arrival of an order in the folder activates an instance of the **NewOrder** orchestration.

The **NewOrder** orchestration first extracts the broker identifier (**SoldToName**) from the order and calls the **GetUserProfile** orchestration to retrieve the profile for the broker. If there is no profile, the orchestration writes an error to the system event log and ends. For more information about the **GetUserProfile** orchestration, see Getting Profile Information in the Business-to-Business Solution .

If a profile exists, the orchestration retrieves the broker's e-mail address from the partner message using a method of the **XmlHelper** object in the **Utilities** assembly. The orchestration next creates an empty order message using the **GetEmptyOrder** method of the **ResourceHelper** object in the Utilities assembly.

The orchestration next sends the order to the back-end system and waits for a response. The back-end order system is a Web service, in this case an orchestration presented as a Web service.

If the **NewOrder** orchestration receives a response, it breaks the retry loop surrounding the Send and Receive shapes. Otherwise, one of the exception blocks handles the exception from the back-end order system.

Determining which block handles an exception in the **NewOrder** orchestration is not simple because there are two layers of indirection. Exceptions come from the **DBHelper** object and are in turn remapped by the back-end system. The back-end order system uses the **PlaceOrder** method of the **DBHelper** object to add the order to the database. The method can throw an application exception or a system exception:

In response to an application exception, the back-end system sends a **BusinessFault** message. This message is sent in response to a null or duplicate order. There is no recovery for these errors so that **NewOrder** terminates itself.

All other exceptions produce a **SystemFault** message. As long as the retry count is less than three, **NewOrder** suspends the order and waits for the operator to intervene.

When the **NewOrder** orchestration receives a confirmation of the order, it calls the **ConfirmOrderWithVendors** and **ConfirmOrderWithBroker** orchestrations. These orchestrations send order information and email notices to the vendors and broker.

## Confirming Orders

The **NewOrder** orchestration confirms orders with both the broker (purchaser) and the vendors. Separate orchestrations handle these confirmations.

The **NewOrder** orchestration passes only the order message to the **ConfirmOrderWithVendors** orchestration. It passes the order message, broker identifier, and the e-mail destination of the broker to the **ConfirmOrderWithBroker** orchestration.

## Confirming with Vendors

The **ConfirmOrderWithVendors** orchestration handles order confirmation with the vendors. It does this by splitting the purchase order into individual orders and sending them to the vendors. If it successfully sends an order, the orchestration then sends a confirming e-mail to the vendor.

The main body of **ConfirmOrderWithVendors** is a loop that runs once for each individual order in the purchase order. The orchestration begins by extracting the number of orders from the purchase order message. It then enters the processing loop.

At the top of the loop, the orchestration extracts an individual order from the purchase order using the order count and an XPath expression. Next, it extracts the vendor identifier from the individual order and calls the **GetUserProfile** orchestration to retrieve the partner information.

If partner information exists, the orchestration sets the destination on the **ContactVendorRoleLink**:

It then sends the order message to the vendor and extracts the vendor's e-mail address from the profile information returned by **GetUserProfile**.

If sending the order succeeds, the orchestration sets the dynamic send port destination for the e-mail notification and sends the e-mail. The orchestration increments the loop counter and goes back for the next order.

Notice that if sending the message to the vendor fails, the exception handler sets a flag so that e-mail is not sent, and creates an event log entry.

## Confirming with the Broker

The structure of the **ConfirmOrderWithBroker** orchestration has the same structure as the code within the loop of the **ConfirmOrderWithVendors** orchestration. And, because the broker identifier and e-mail destination are arguments to it, the orchestration is quite short. It sets the **BrokerRoleLink** destination and sends the order message. Similarly, it sets the destination for the **SendBrokerEmail** port and then sends the e-mail.

Exception handling follows the same pattern as in the **ConfirmOrderWithVendors** orchestration—if sending the order message fails, the e-mail is not sent.

## Getting Profile Information in the Business-to-Business Solution

The solution uses the **GetUserProfile** orchestration to retrieve trading partner information from the Trading Partner Management Web service. The solution wraps the standard Web service with its own Web service, **TPMAccess**, in order to simplify the calls. In particular, **TPMAccess** handles the credentials required for calls into the partner management system.

For more information about trading partner management and the standard Web service, see Management and Tracking Architecture, Managing Partner Relationships with BAS, and TPPubWS. For information about **TPMAccess**, open the file **TPMAccess.asmx.cs** in Microsoft Visual Studio 2005 and see "The TPMAccess Web Service" in Trading Partner Management in the Business-to-Business Solution.

The **GetUserProfile** orchestration takes three arguments: *userID*, a string containing the identifier for the partner; *bIntervention*, a Boolean set to indicate whether the profile is found and whether operator intervention is needed; and *ReturnProfile*, a message using the **PartnerObject** schema that contains the profile information.

The **GetUserProfile** orchestration begins by assigning the *userID* parameter to the *PartnerID* of the query message and constructing an empty message to return the result.

Next, the orchestration sends the query message to the **TPMAccess** Web service and waits for a reply. If there is a reply, it assigns the returned information to the result message. The partner information comes from the Web service as a string. The orchestration uses the **ConvertStringToXml** method of the **XmlHelper** object in the **Utilities** assembly to convert the string. Also, if there is a reply, the orchestration sets the *bIntervention* argument to false and sets *retryCount* to three to break out of the retry loop.

If there is an exception receiving a reply and there have been fewer than three tries, the orchestration logs a suspension and suspends the orchestration. Otherwise, it logs an error and sets the intervention flag, *bIntervention*, to true.

## Retrieving Orders in the Business-to-Business Solution

The **GetOrder** orchestration allows any partner enlisted in the **GetOrderLinkType** role link to check order status.

## GetOrder

The **GetOrder** orchestration takes an **OrderInquiryMessage**; retrieves the order identifier, partner name, and broker identifier; and calls the **GetPurchaseOrder** orchestration. If the **GetPurchaseOrder** orchestration finds a corresponding order (the orchestration returns a value of false for the variable *bFailure*), the **GetOrder** orchestration constructs and sends an **Order** message. Otherwise, **GetOrder** creates an **OrderAck** message with a result of false, and sends that message.

## GetPurchaseOrder

The **GetPurchaseOrder** orchestration performs the query of the back-end order system. It takes four parameters: *brokerID*, the identifier for the broker; *orderID*, the order identifier; *OrderMessage*, a message returned containing the order queried; and *bFailure*, a Boolean variable set to true if the order query fails.

The orchestration creates an empty order message using the **GetEmptyOrderMessage** method of the **ResourceHelper** object in the Utilities assembly. Next, the orchestration converts the order message into an order query using a map, and copies the *brokerID* and *orderID* values into the query message.

The orchestration sends the request to the back-end order system and waits for a response. If there is a response, the orchestration sets the retry counter to a value greater than three to break out of the retry loop, and exits.

If the request fails, the back-end order system returns an exception. No matter what kind of exception occurs, the orchestration logs the error and suspends the process if the retry count is less than three. Otherwise, it makes an entry in the system log, sets the *bFailure* flag to true to indicate that intervention is needed, and returns to the calling orchestration.

# Updating Orders in the Business-to-Business Solution

## The UpdateOrder Orchestration

When the **UpdateOrder** orchestration receives a message through the POChangeRoleLink, it verifies the sender by extracting the partner name from the role link and the sold-to name from the updated order. If the two do not match, the sender is invalid. The orchestration makes an entry in the event log and ends.

If the sender is valid, the orchestration uses a map to convert the received purchase order into an order query. The orchestration calls the **CheckOrderExists** orchestration to ensure that the order being updated exists. If the order doesn't exist, the **CheckOrderExists** orchestration makes an entry in the event log and returns to the **UpdateOrder** orchestration.

If the order exists, **UpdateOrder** sends the purchase order, using the Web service, to the back-end order system, and waits for a response. This section of the orchestration is another suspend-retry loop and functions exactly the same as the loop in the **NewOrder** orchestration. For a description of the pattern and actions, see Processing Orders in the Business-to-Business Solution.

If the update succeeds, **UpdateOrder** calls the **ConfirmOrderWithVendors** orchestration. It then retrieves the broker's e-mail address from the partner profile (retrieved by a call to **GetUserProfile**) and calls the **ConfirmOrderWithBroker** orchestration. For more information about the **ConfirmOrderWithVendors** and **ConfirmOrderWithBroker** orchestrations, see "Confirming Orders" in Processing Orders in the Business-to-Business Solution. For information about the **GetUserProfile** orchestration, see Getting Profile Information in the Business-to-Business Solution .

## Retrieving Partner Profiles in the Business-to-Business Solution

The **GetProfile** orchestration enables anyone enlisted in the **GetProfileRoleLink** role link to retrieve profile information. The orchestration extracts the partner name from the role link and calls the **GetUserProfile** orchestration. For information about the **GetUserProfile** orchestration, see Getting Profile Information in the Business-to-Business Solution .

**GetProfile** then checks the flag returned by **GetUserProfile** to see if the profile was found. If the profile is not found, **GetProfile** creates an empty **ProfileAck** message using the **GetEmptyProfileAck** method of the **ResourceHelper** object in the **Utilities** assembly. **GetProfile** assigns the user identifier to the **PartnerName** message field, and false to the **Result** message field, and sends the message using the **ProfileStatusPort** port. For information about creating empty messages, see Constructing Messages in User Code .

If **GetUserProfile** finds the profile, **GetProfile** sends the profile back using the **ProfilePort** port.

## Updating Partner Profiles in the Business-to-Business Solution

The **UpdateUserProfileBAS** orchestration lets anyone enlisted in the **UpdateProfileRoleLink** role link to update their profile.

The orchestration takes a message using the **PartnerObject** schema. The orchestration extracts the partner identifier from the message and tests it against the partner identifier from the role link. If the two do not match, the orchestration writes an error to the system event log and returns.

If the identifiers match, the orchestration constructs an **UpdateProfileRequest** message for the **TPMAccess** Web service. Recall that this is the Web service the solution uses to wrap the **TPPubWS** Web service. The orchestration converts the message to a string using the **ConvertXmlToString** method of the **XmlHelper** object in the **Utilities** assembly.

The orchestration sends the message to the Web service. If there is an error, an exception handler catches the mistake, logs the event, and suspends the message. After the third retry, the orchestration sends a **ProfileAck** message with *Result* set to false. The orchestration reports success or failure to the response port of the **UpdateProfileRoleLink**.

For more information about the TPM Web services, see TPPubWS. For more information about the **PartnerObject**, see About Partner Schemas .



## Using EDI in the Business-to-Business Solution

The Business-to-Business solution can also take orders using EDI. The solution uses an orchestration, **EDIXmlToBasket**, as a front-end to the **NewOrder** orchestration. **EDIXmlToBasket** receives the EDI order, converts it to a Commerce Server basket, and sends it on to the **NewOrder** orchestration.

To handle both EDI and file orders, the **NewOrder** orchestration has two receive locations assigned to its receive port. The receive port in the orchestration, **RecvBasketPort**, maps to the physical port **NewOrderPickupRP**. This port includes two receive locations, **OrderPickup\_FILE** (c:\litware\BrokerNew) and **OrderPickupEDI\_FILE** (c:\Litware\EDIOut). A message in either receive location activates the port and the **NewOrder** orchestration.

When **EDIXmlToBasket** receives an order, it first uses the **GetPartnerID** method of the **PartnerHelper** object in the **Utilities** assembly to get the broker identifier (*brokerID*) and the partner identifier (*partnerID*). The **GetPartnerID** method does a database lookup of the receive location to translate it into the correct identifier. For the *brokerID*, it uses the **InboundTransportLocation** of the message; and for *partnerID*, the **DestinationParty**.

**EDIXmlToBasket** next uses a map to convert the EDI message into a Commerce Server basket message. It then assigns the *brokerID* to the **SoldToName** promoted property and assigns the *partnerID* to the name in the order form using an XPath expression. Finally, it sends the finished message to the file location where the **NewOrder** orchestration will pick it up.

For more information about the EDI adapter, see *Configuring the Base EDI Adapter and Base EDI Adapter*.

## Implementation Highlights of the Business-to-Business Solution

This section describes some elements of the Business-to-Business solution in greater detail. These elements include how the solution uses the Trading Partner Management system, the Party Resolution pipeline component, and some of the helper methods and objects.

### In This Section

Trading Partner Management in the Business-to-Business Solution

Message Brokering in the Business-to-Business Solution

## Trading Partner Management in the Business-to-Business Solution

Because the solution relies on the Trading Partner Management system for routing all of the orders, confirmations, and email messages, you must configure the system properly for the solution to operate. In a production environment, the centralized routing of the message broker pattern requires regular maintenance to ensure the routing information is current.

The solution includes several Microsoft® InfoPath™ forms to use to maintain the partner information.

## Extending the Trading Partner Management System

The Business-to-Business solution extends the Trading Partner Management system to accommodate items needed to work with Commerce Server. You can extend the Trading Partner system following the same process. You need to do three things:

Add the necessary columns to the BizTalk database.

Revise the profile schemas to include the new information.

Update the InfoPath forms.

For step-by-step instructions about extending the trading partner management system, see Extending Partner Schemas and How to Install the Business-to-Business Solution.

## Adding Columns

You add columns for new profile items to the **ContactObject** table in the **BizTalkMgmtDb** database. The Business-to-Business Solution uses the following SQL to add three items to the database:

For more information about running the SQL, see "To extend the BAS profile" in How to Install the Business-to-Business Solution.

## Revising the Profile Schemas

There are two places in the profile schema that you need to enter information for each new item.

The first is to add a **Property name** entry in the **Profile name** section for the **ContactObject**. Notice in the entry for **UserName** in TPMUserSchema.xml that you also include a reference to the database column:

For information about the custom properties, see "Profile Property Attributes" in the Microsoft Commerce Server help or <http://go.microsoft.com/fwlink/?linkid=55415>.

The second place to add information is in the **DataSource** section to which you add the new database columns in the **DataObject** section for the **ContactObject**. For example, the solution uses the following code:

Noitice that these entries simply describe the database columns.

## Updating the InfoPath Forms

For guidance on updating the InfoPath forms for the new profile items, see the forms in the solution code directory.

## The TPMAccess Web Service

The TPMAccess Web service provides two methods, GetPartner for retrieving partner profile information, and UpdatePartner for updating trading partner information. Both methods take a single string argument:

TPMAccess Method	Argument	Description
GetPartner	partnerID (string)	Takes the partner identifier in the argument and gets the partner profile.
UpdatePartner	partnerXML (string)	Takes the string containing an XML document following the PartnerObject record format of the Partner schema.

Notice that when used in code, the string passed to the UpdatePartner method is the value returned by the ConvertXMLToString helper method. This is necessary to convert the record (partial XML document) into a complete XML document before passing it on for updating the trading partner management system.

## Message Brokering in the Business-to-Business Solution

The message broker pattern represents a centralization of the routing of messages in a system. This is particularly true in this implementation where the routing information is kept in the partner system. In production, the solution would need to be monitored in order to ensure that the centralized routing does not become a processing bottleneck.

## Certificates and Signing Messages

The business-to-business solution supports signed and unsigned messages. This section outlines how to sign messages and how to create certificates for signing.

### Creating Signed Messages

The scenario accepts unsigned XML and EDI flat file messages so that you can use the scenario without signed messages. It also supports receiving signed messages for profile updates and order status inquiries. To use this feature, you need to create S/MIME signed messages based on the unsigned samples OrderInquiry.xml and Partner.xml SDK\Scenarios\B2B\SampleMessages directory.

You can create signed messages several ways:

In Outlook Express you can click on the sign icon and have the message signed if you have a certificate. This is the same certificate you use in the BizTalk Server Trading Partner Management system for the partner you are simulating.

You need to modify the solution to use the POP3 adapter. You also need to use the MIME/SMIME Decoder pipeline component. This component processes the digital signature and sender, and puts the sender information in the message header.

You can write a simple routing application using BizTalk Server. The application can take messages as XML files using the file adapter, sign them, and place them in another directory as signed messages. BizTalk supports one certificate per BizTalk group. If you want to simulate more than one partner, you need to create additional BizTalk groups for each additional partner that you would like to simulate. Or, you need to change the certificate and run the messages through again to create new, signed messages for the next trading partner.

## Creating Certificates

To create and install a certificate, you use the Visual Studio command line tool, makecert.exe.

It is possible that you have multiple versions of makecert.exe on your computer. Specify the full path when using the tool to ensure you're using the correct version.

Before using the tool, also be sure that you are logged in with the same user account that the BizTalk service uses.

You want to generate a public/private key pair and have them installed appropriately. For example, to create a key pair with a common name of www.myserver.com, you would type the following command:

```
makecert -r -pe -n "CN=www.myserver.com" -b 01/01/2000 -e 01/01/2036 -eku 1.3.6.1.5.5.7.3.1 -ss my -sr localMachine -sky exchange -sp "Microsoft RSA SChannel Cryptographic Provider" -sy 12
```

For information about all of the options used with makecert.exe, see "Certificate Creation Tool (Makecert.exe)" in .Net Framework Tools.

## Business-to-Business Solution Reference

This section provides lists of the files and messages in the solution.

### In This Section

File Inventory for the Business-to-Business Solution

Message Reference for the Business-to-Business Solution

## File Inventory for the Business-to-Business Solution

DIRECTORY: .

File	Description
Cleanup.bat	Description
Microsoft.Samples.BizTalk.Litware.sln	Description

Readme.html	Description
ReplacePKToken.vbs	Description
ReplacePKToken.wsf	Description
Setup.bat	Description

DIRECTORY: ../Bindings

File	Description
BindingOrchestration.xml	Description
BindingOrderSystem.xml	Description
OtherPorts.xml	Description
PartyPorts.xml	Description

DIRECTORY: ../Maps

File	Description
Basket_To_PurchaseOrder.btm	Description
Maps.btmproj	Description
PurchaseOrder_To_Inquiry.btm	Description
PurchaseOrder_To_OrderAck.btm	Description
X12_4010_850_To_Basket.btm	Description

DIRECTORY: ../Orchestrations

File	Description
CheckOrderExists.odx	Description
ConfirmOrderWithBroker.odx	Description
ConfirmOrderWithVendors.odx	Description

EDIXmlToBasket.odx	Description
GetOrder.odx	Description
GetProfile.odx	Description
GetPurchaseOrder.odx	Description
GetUserProfile.odx	Description
NewOrder.odx	Description
Orchestrations.btproj	Description
UpdateOrder.odx	Description
UpdateUserProfileBAS.odx	Description

DIRECTORY: ../Orchestrations/Web References/TPMAccess

File	Description
Reference.map	Description
Reference.odx	Description
TPMAccess.disco	Description
TPMAccess.html	Description
TPMAccess.wsdl	Description

DIRECTORY: ../OrderSystem

File	Description
GetOrder.odx	Description
IsOrderExist.odx	Description
NewOrder.odx	Description
OrderSystem.btproj	Description

UpdateOrder.odx	Description
-----------------	-------------

DIRECTORY: ./OrderSystem/SQL

File	Description
Order.SQL	Description

DIRECTORY: ./PartyResolutionComponent

File	Description
AssemblyInfo.cs	Description
PartyResolutionComponent.cs	Description
PartyResolutionComponent.csproj	Description
PartyResolutionStream.cs	Description

DIRECTORY: ./Pipelines

File	Description
PartnerReceive.btp	Description
PartnerSend.btp	Description
Pipelines.btproj	Description
UpdateOrderSoldToNamePR.btp	Description

DIRECTORY: ./SampleMessages

File	Description
Basket.XML	Description
OrderInquiry.xml	Description
Partner.XML	Description
PO.XML	Description

RequestProfile.xml	Description
--------------------	-------------

DIRECTORY: ../Schemas

File	Description
PropertySchema.xsd	Description
Schemas.btproj	Description

DIRECTORY: ../Schemas/EDI

File	Description
X124010850Schema.xsd	Description

DIRECTORY: ../Schemas/Order

File	Description
Basket.xsd	Description
OrderForm.xsd	Description
OrderTemplate.xsd	Description
PurchaseOrder.xsd	Description
RequestOrder.xsd	Description

DIRECTORY: ../Schemas/OrderSystem

File	Description
BusinessFaultMessage.xsd	Description
OrderAck.xsd	Description
OrderInquiry.xsd	Description
SystemFaultMessage.xsd	Description

DIRECTORY: ../Schemas/TPM



File	Description
FileBinding.xsd	Description
HttpBinding.xsd	Description
MessageProcessing.xsd	Description
Partner.xsd	Description
ProfileAck.xsd	Description
RequestProfile.xsd	Description
ServiceLinkType.xsd	Description
SoapBinding.xsd	Description
wsdl.xsd	Description
WssBinding.xsd	Description

DIRECTORY: ../Scripts

File	Description
CreateLitwareApplication.cmd	Description
LitwareSSOConfiguration.xml	Description
Locations.cmd	Description

DIRECTORY: ../TPM/Certificate

File	Description
myserver.pfx	Description

DIRECTORY: ../TPM/InfoPathform

File	Description
PartnerProfileAdmin.xsn	Description
PartnerProfileSelf.xsn	Description

PartnerProfileStandard.xsn	Description
----------------------------	-------------

DIRECTORY: ../TPM/PartnerData

File	Description
Agreement.xml	Description
Agreement2.xml	Description
MyProfile.xml	Description
TestPartner2Extended.xml	Description
TestPartnerExtended.xml	Description

DIRECTORY: ../TPM/Schemas

File	Description
Partner.xsd	Description
PartnerNoWSDL.xsd	Description

DIRECTORY: ../TPM/SQL

File	Description
Contact.SQL	Description

DIRECTORY: ../TPM/TPMSchema

File	Description
TPMUserSchema.XML	Description
TPMUserSchemaNotEncrypted.XML	Description

DIRECTORY: ../TPMAccessWS

File	Description
Global.asax	Description

Global.asax.resx	Description
Microsoft.Samples.BizTalk.Litware.TPMAccessWS.sln	Description
TPMAccess.asmx	Description
TPMAccess.asmx.resx	Description
Web.config	Description

DIRECTORY: ../TPMAccessWS/App\_Code

File	Description
AssemblyInfo.cs	Description
Global.asax.cs	Description
TPMAccess.asmx.cs	Description
TPPubWS.cs	Description

DIRECTORY: ../TPMAccessWS/App\_Data

File	Description
TPPubWS.wsdl	Description

DIRECTORY: ../Utilities

File	Description
AssemblyInfo.cs	Description
DBHelper.cs	Description
Helper.cs	Description
PartnerHelper.cs	Description
Resources.cs	Description
Resources.resx	Description

SSOConfigHelper.cs	Description
Utilities.csproj	Description

## Message Reference for the Business-to-Business Solution

This section lists the messages and message types used by each orchestration in the solution.

In the table, <SolutionSchemas> replaces Microsoft.Samples.BizTalk.Litware.Schemas to help the table's formatting. Similarly, <SolutionOrchestrations> replaces Microsoft.Samples.BizTalk.Litware.Orchestrations.

The orchestrations, messages, and types are as follows.

Orchestration	Message	Message Type
CheckOrderExists.odx	InquiryResponseMsg	<SolutionSchemas>.OrderSystem.OrderInquiry
CheckOrderExists.odx	InquiryMsg	<SolutionSchemas>.OrderSystem.OrderInquiry
ConfirmOrderWithBroker.odx	OrderMessage	<SolutionSchemas>.Order.PurchaseOrder
ConfirmOrderWithVendors.odx	OrderFormMsg	<SolutionSchemas>.Order.OrderForm
ConfirmOrderWithVendors.odx	ProfileMsg	<SolutionSchemas>.TPM.Partner.PartnerObject
ConfirmOrderWithVendors.odx	OrderMessage	<SolutionSchemas>.Order.PurchaseOrder
EDIXmlToBasket.odx	Source	<SolutionSchemas>.EDI.X124010850Schema
EDIXmlToBasket.odx	Target	<SolutionSchemas>.Order.Basket
GetOrder.odx	OrderMessage	<SolutionSchemas>.Order.PurchaseOrder
GetOrder.odx	OrderInquiryMessage	<SolutionSchemas>.OrderSystem.OrderInquiry
GetOrder.odx	OrderAck	<SolutionSchemas>.OrderSystem.OrderAck
GetProfile.odx	INMessage	<SolutionSchemas>.TPM.RequestProfile

GetProfile.odx	CALLMessage	<SolutionSchemas>.TPM.Partner.PartnerObject
GetProfile.odx	ProfileAck	<SolutionSchemas>.TPM.ProfileAck
GetPurchaseOrder.odx	RequestMessage	<SolutionSchemas>.OrderSystem.OrderInquiry
GetPurchaseOrder.odx	OrderMessage	<SolutionSchemas>.Order.PurchaseOrder
GetUserProfile.odx	QueryMessage	<SolutionOrchestrations>.TPMAccess.TPMAccess_.GetP
GetUserProfile.odx	Profile	<SolutionOrchestrations>.TPMAccess.TPMAccess_.GetP
GetUserProfile.odx	ReturnProfile	<SolutionSchemas>.TPM.Partner.PartnerObject
IsOrderExist.odx	InquiryMsg	<SolutionSchemas>.OrderSystem.OrderInquiry
IsOrderExist.odx	ResponseMsg	<SolutionSchemas>.OrderSystem.OrderInquiry
IsOrderExist.odx	SystemFaultMessage	<SolutionSchemas>.OrderSystem.SystemFaultMessage
UpdateOrder.odx	POChgMsg	<SolutionSchemas>.Order.PurchaseOrder
UpdateOrder.odx	InquiryMsg	<SolutionSchemas>.OrderSystem.OrderInquiry
UpdateOrder.odx	UpdateAck	<SolutionSchemas>.OrderSystem.OrderAck
UpdateOrder.odx	ProfileMsg	<SolutionSchemas>.TPM.Partner.PartnerObject
UpdateUserProfileBAS.odx	ProfileMessage	<SolutionSchemas>.TPM.Partner.PartnerObject
UpdateUserProfileBAS.odx	UpdateProfileRequest	<SolutionOrchestrations>.TPMAccess.TPMAccess_.Upd
UpdateUserProfileBAS.odx	UpdateProfileResponse	<SolutionOrchestrations>.TPMAccess.TPMAccess_.Upd
UpdateUserProfileBAS.odx	ProfileAck	<SolutionSchemas>.TPM.ProfileAck

The **OrderSystem** orchestrations have a separate set of messages described in the following table. The table uses the same abbreviations as the previous table.

Orchestration	Message	Message Type
GetOrder.odx	SystemFaultMessage	<SolutionSchemas>.OrderSystem.SystemFaultMessage
GetOrder.odx	InquiryMsg	<SolutionSchemas>.OrderSystem.OrderInquiry

GetOrder.odx	OrderMessage	<SolutionSchemas>.Order.PurchaseOrder
IsOrderExist.odx	InquiryMsg	<SolutionSchemas>.OrderSystem.OrderInquiry
IsOrderExist.odx	ResponseMsg	<SolutionSchemas>.OrderSystem.OrderInquiry
IsOrderExist.odx	SystemFaultMessage	<SolutionSchemas>.OrderSystem.SystemFaultMessage
NewOrder.odx	BasketMessage	<SolutionSchemas>.Order.Basket
NewOrder.odx	OrderMessage	<SolutionSchemas>.Order.PurchaseOrder
NewOrder.odx	SystemFaultMessage	<SolutionSchemas>.OrderSystem.SystemFaultMessage
NewOrder.odx	BusinessFaultMessage	<SolutionSchemas>.OrderSystem.BusinessFaultMessage
UpdateOrder.odx	OrderMessage	<SolutionSchemas>.Order.PurchaseOrder
UpdateOrder.odx	OrderAckMessage	<SolutionSchemas>.OrderSystem.OrderAck
UpdateOrder.odx	BusinessFaultMessage	<SolutionSchemas>.OrderSystem.BusinessFaultMessage
UpdateOrder.odx	SystemFaultMessage	<SolutionSchemas>.OrderSystem.SystemFaultMessage

### Before Installing the Business-to-Business Solution

The following prerequisites must be available in order to install the Business-to-Business solution on a single computer:

Microsoft BizTalk Server 2006 with the following components:

- Base EDI Adapter

- Portal Components

- Windows SharePoint Services Adapter Web Service

Microsoft SQL Server 2000 Service Pack 4

Microsoft Internet Information Services 6.0 with the following services:

- World Wide Web Publishing Service

- SMTP Services

ASP.NET 2.0 enabled

Microsoft Visual Studio 2005

Microsoft InfoPath 2003 with Service Pack 2

## How to Install the Business-to-Business Solution

The following steps describe how to prepare the computer for installing the Business-To-Business (B2B) solution, and how to install the solution on this computer.

Prepare the computer for installing the Business-to-Business Solution

Extend the BAS profile

Set certificates for processing signed messages

Compile the Business-to-Business Solution

Register BizTalk Server on the BAS site

Configure the hosts as Authentication Trusted

Install the Business-to-Business Solution

Create the profile using the BAS site

Create the trading partner using the BAS site

Deploy the Business-to-Business Solution

### Prepare the computer for installing the Business-to-Business Solution

#### To prepare the computer for installing the Business-to-Business Solution

Click **Start**, point to **All Programs**, point to **Administrative Tools**, click **Computer Management**, and then make sure that the BizTalk service account running SharePoint Adapter is a member of the local SharePoint Enabled Hosts Windows group on the server. For this walkthrough, use the same BizTalk service account for all of the services. For more information about Windows SharePoint Services Adapter, see What Is the Windows SharePoint Services Adapter?

In the **Computer Management** console, make sure that the HWS Web Service user is member of the local IIS\_WPG and STS\_WPG windows groups. For this walkthrough, use the same BizTalk service account for all of the services.

Exclude the (root) of the Default Web Site from Windows SharePoint Services Managed Paths as follows: Click **Start**, point to **All Programs**, point to **Administrative Tools**, and then click **SharePoint Central Administration**.

Under **Virtual Server Configuration**, select **Configure virtual server settings**.

On the **Virtual Server List** page, click **Default Web Site**.

On the **Virtual Server Settings** page, click **Define managed paths**.

In the **Included Paths** section of the **Defined Managed Path** page, select **Root** and then click **Remove selected paths**.

At a command prompt, perform an IISReset.

In the **Computer Management** console, add the BizTalk service account to the local BizTalk BAS Web Services Group, BizTalk BAS Administrators, BizTalk BAS Managers, BizTalk BAS Users groups, and local Administrators group.

At a command prompt, change the directory to the %ProgramFiles%\Common Files\Microsoft Shared\web server extensions\60\BIN, type the following command, and then press ENTER.

```
stsadm.exe -o upgrade -forceupgrade -url http://localhost
```

Log off the computer, and then log on to the computer as the BizTalk service account.

At a command prompt, type the following command, and then press ENTER to set the %BTSSolutionsPath% environment variable to indicate the base folder for the E2E solutions. Then, exit the command prompt.

```
setx      BTSSolutionsPath      "%ProgramFiles%\Microsoft      BizTalk      Server  
2006\SDK\Scenarios"
```

## **Extend the BAS profile**

### **To extend the BAS profile**

Click **Start**, point to **All Programs**, point to **Microsoft SQL Server**, and then click **Query Analyzer**.

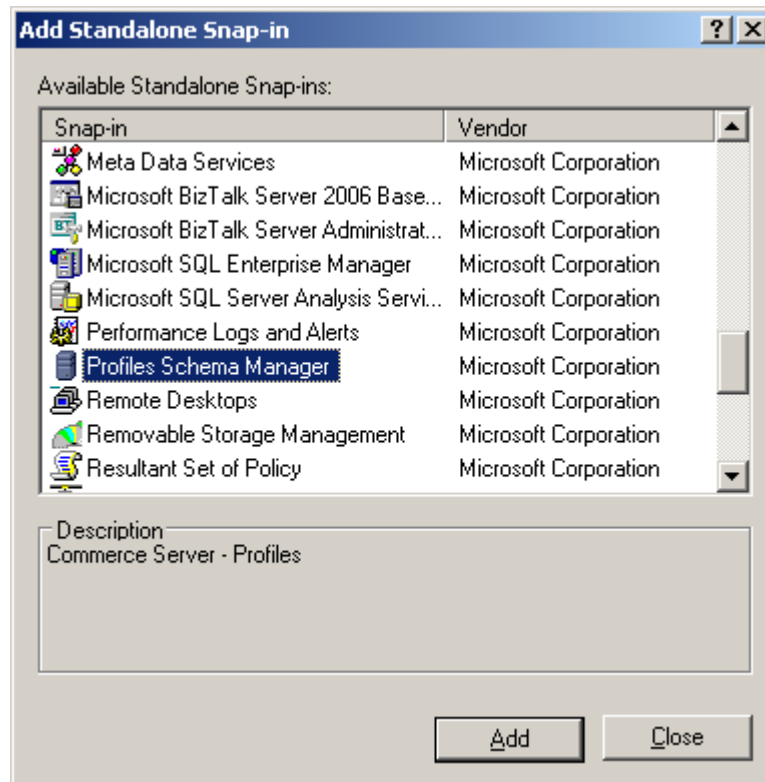
On the **Connect to SQL Server** dialog box, type <Your Computer Name> in the **SQL Server** text box, select **Windows authentication**, and then click **OK**.

Open the Contact.SQL in the %BTSSolutionsPath%\B2B\TPM\SQL folder, and then run the SQL query against the TPM database. The script adds three more columns (UserName, Password, keyindex) to the ContactObject table.

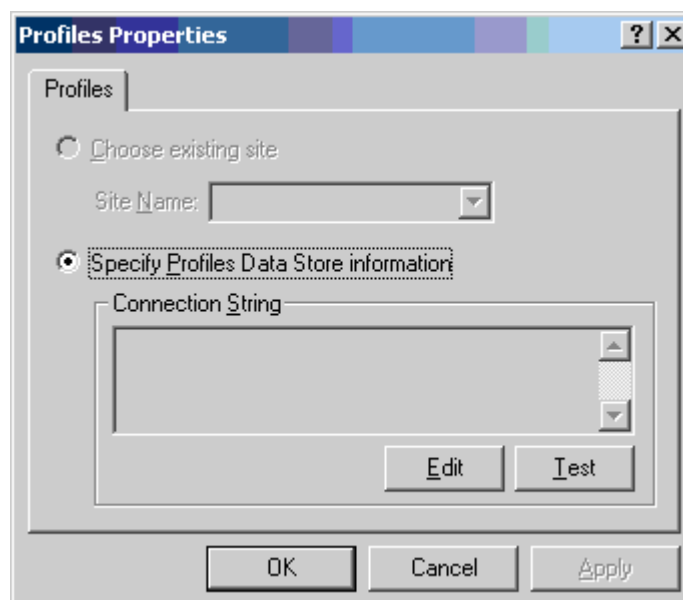
Click **Start**, click **Run**, type **MMC.exe**, and then press ENTER.

In the Microsoft Management Console, add the **Profiles Schema Manager** snap-in.

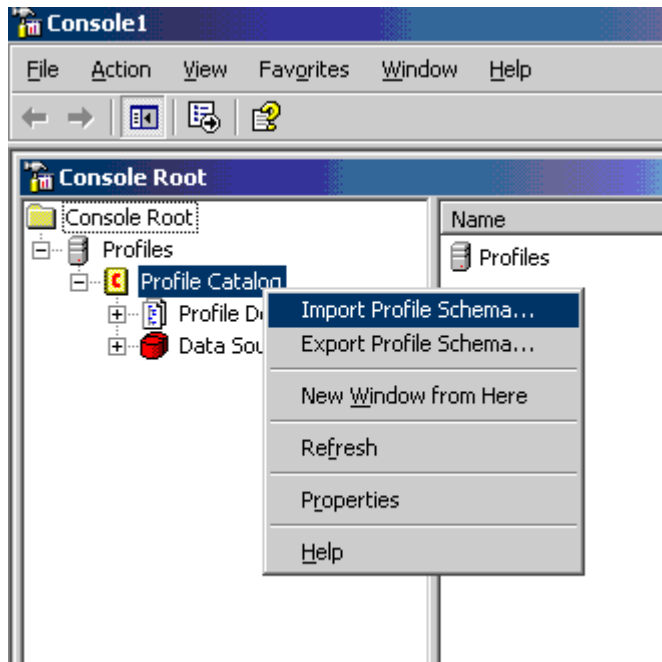




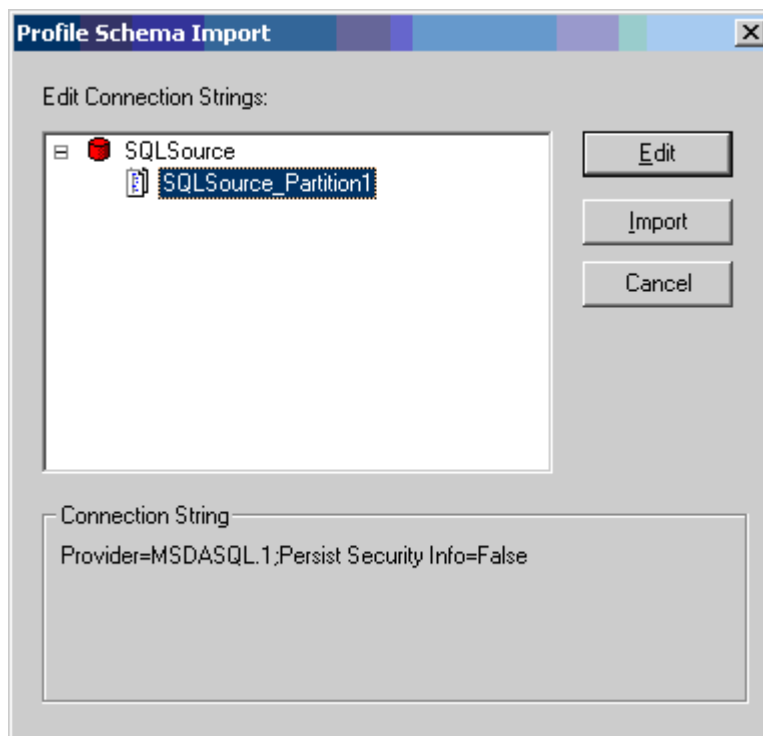
Expand **Profiles**, right-click **Profiles**, select **Properties**, click **Edit** and then set your SQL connection properties to point to your TPM database. Then close **Data Link Properties** and **Profile Properties** dialog boxes.



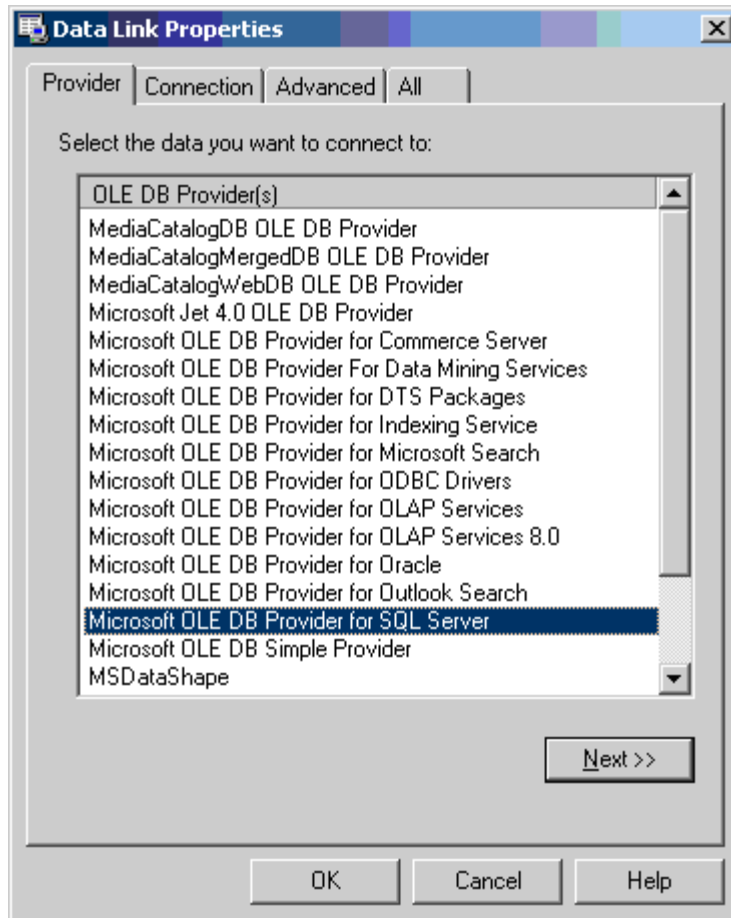
Right-click **Profiles**, click **Import Profile Schema**, select TPMUserSchemaNotEncrypted.xml file in the %BTSSolutionsPath%\B2B\TPM\TPMSchema folder, and then click **OK**.



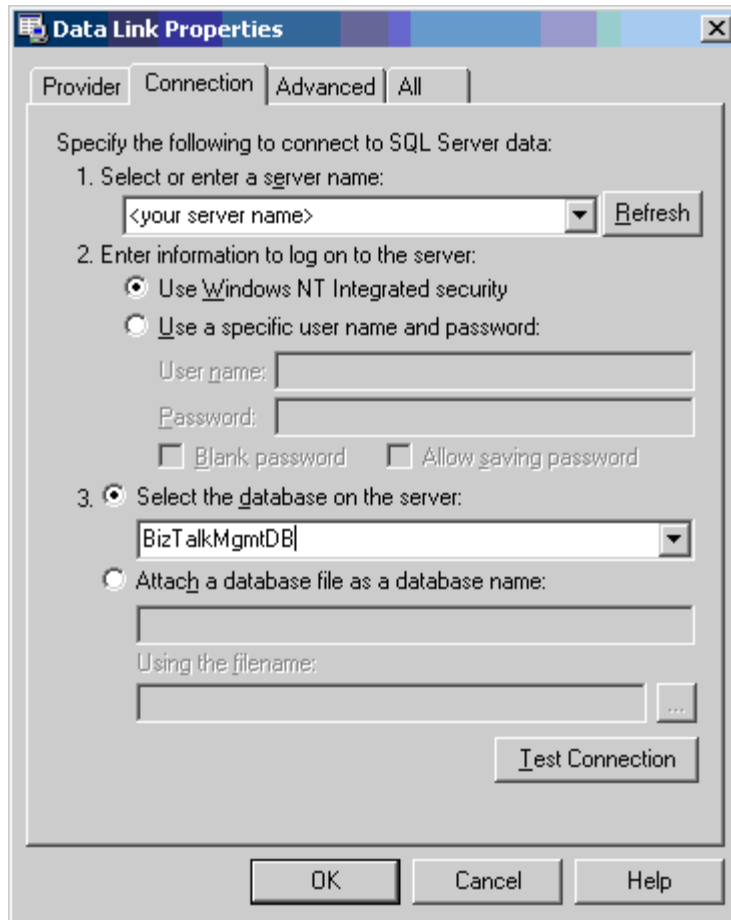
On the **Profile Schema Import** dialog box, expand **SQL Source**, select **SQLSource\_Partion1**, and then click **Edit**.



On the **Data Link Properties** dialog box, click **Providers**, select **OLEDB Provider used is for SQL Server**.



On the **Data Link Properties** dialog box, click **Connection**, select <**Your Computer Name**> in the **Select or enter a server name** box, select **Use Windows NT Integrated security**, select BizTalkMgmtDB in the **Select the database on the server** box, and then click **OK**.



On the **Profile Schema Import** dialog box, click **Import** to import the extended profile schema, and then click **Yes** on the confirmation dialog box.

Expand **Profile Catalog**, expand **Data Sources**, expand **SQLSource**, expand **Data Objects** to review the extend schemas.

Click **Start**, point to **All Programs**, point to **Microsoft BizTalk Server 2006**, and then click **Business Activity Services Site**.

On the **Home** page, click **Documents and Lists**, and then click **Templates** on the **Documents and Lists** page.

On the **Templates** page, click **Upload Document**, and then upload the PartnerProfileAdmin.xsn in the %BTSSolutionsPath%\B2B\TPM\InfoPathform folder.

Repeat the same steps with the PartnerProfileAdmin.xsn for uploading PartnerProfileSelf.xsn and PartnerProfileStandard.xsn in the %BTSSolutionsPath%\B2B\TPM\InfoPathform folder.

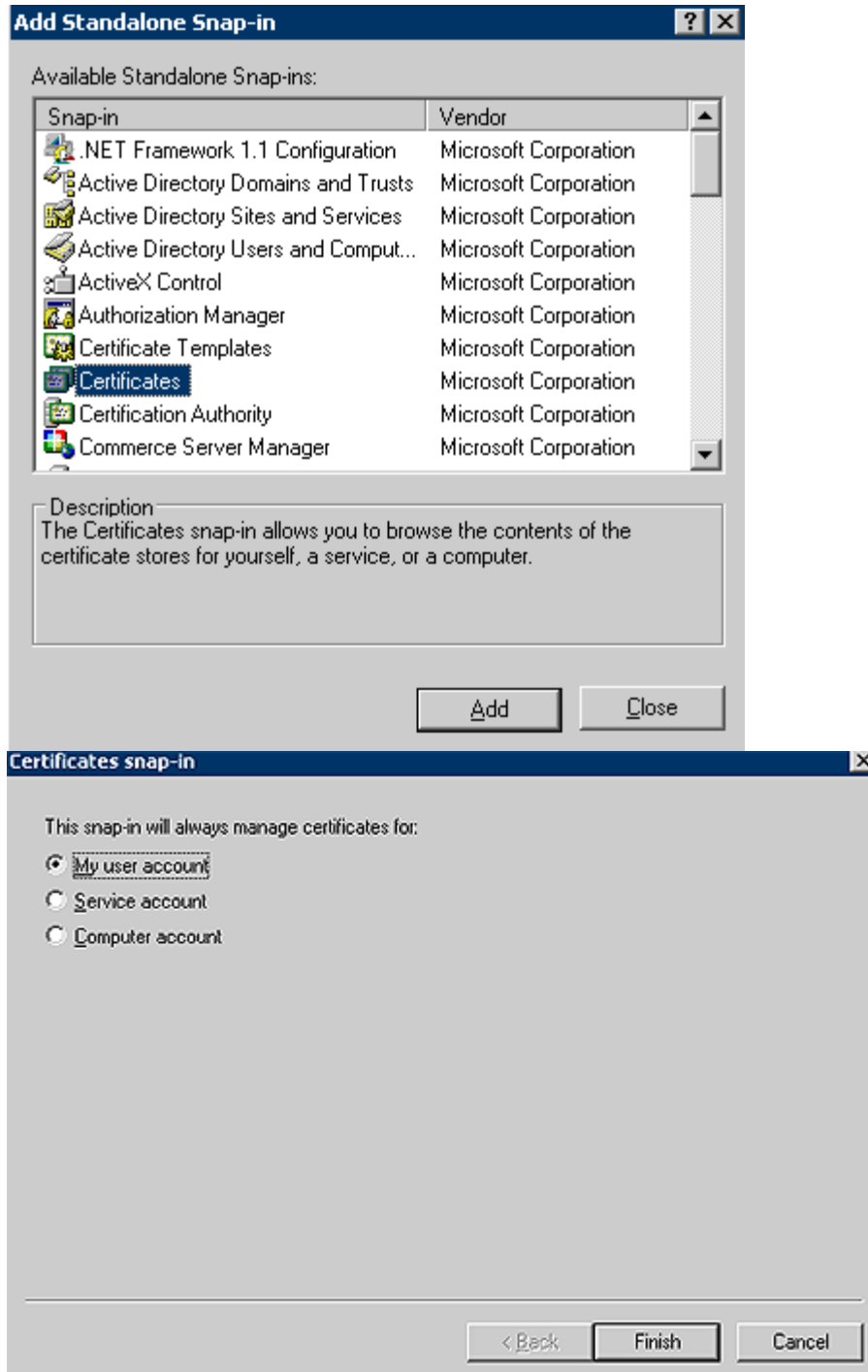
At a command prompt change the directory to the %BTSSolutonsPath%\B2B\TPM\Schemas folder, copy the Partner.xsd and PartnerNoWsdI.xsd files to the <BizTalk Install Directory>\Business Activity Services\TPM\Schemas folder.

## Set certificates for processing signed messages

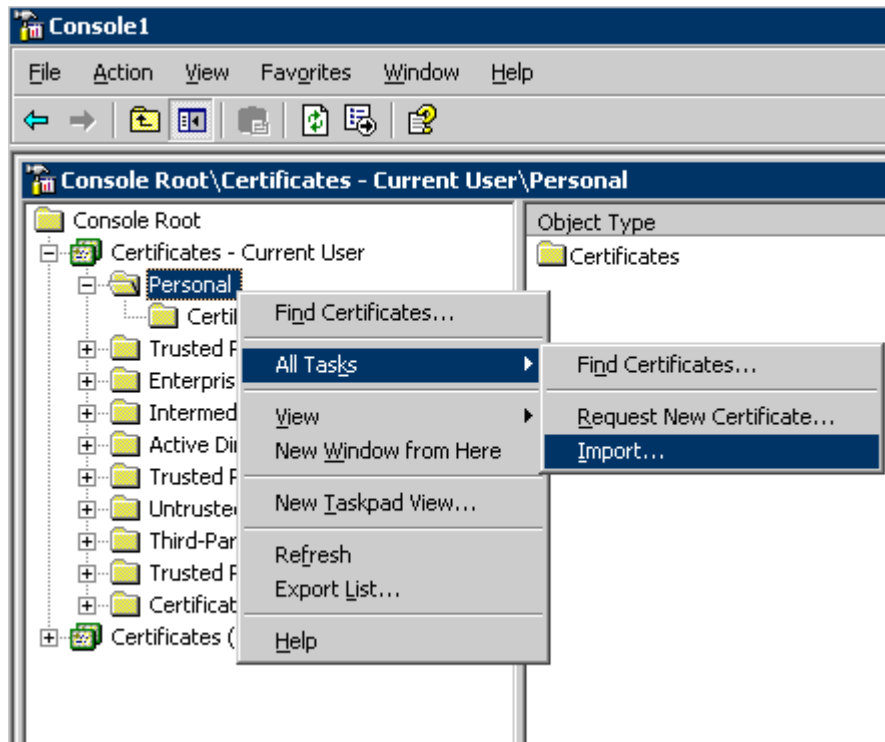
### To set certificates for processing signed messages

Click **Start**, click **Run**, type **MMC.exe**, and then press ENTER.

In the Microsoft Management Console, add the **Certificates** snap-in for **My user account**.

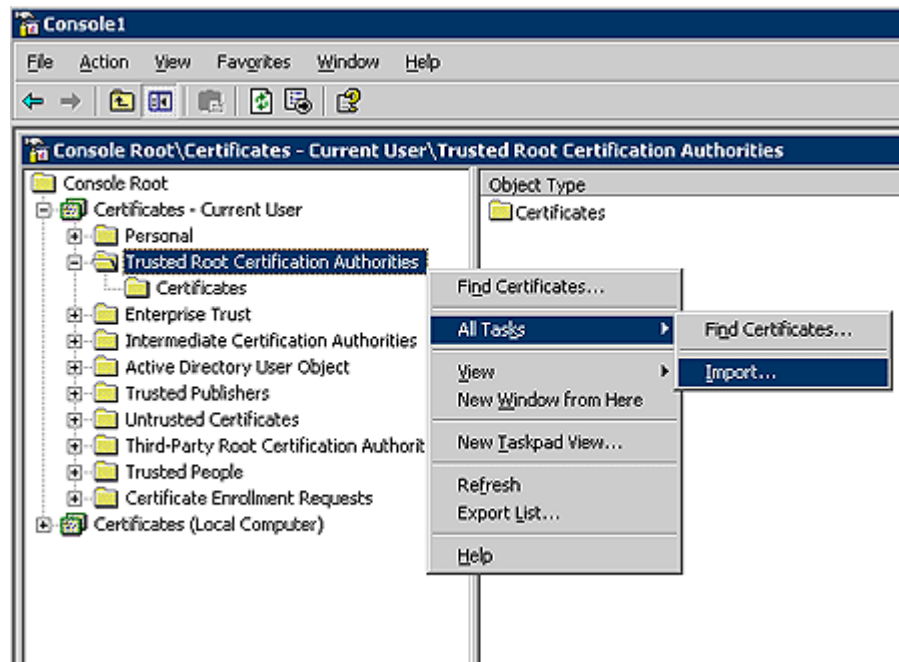


In the **Certificates** snap-in, expand **Certificates - Current User**, right-click **Personal**, point to **All Tasks**, and then click **Import**.



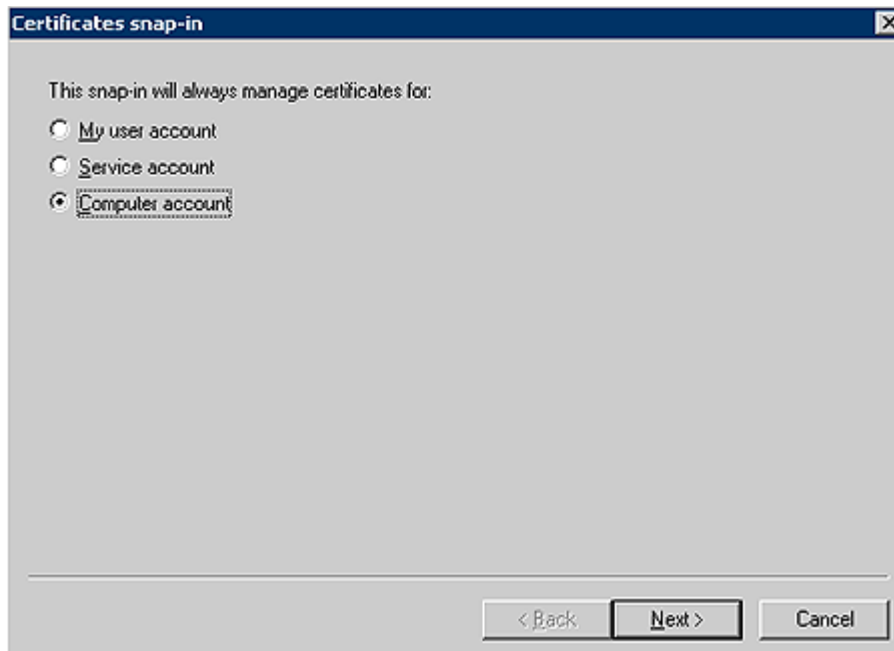
Using the **Certificate Import Wizard**, import the **myserver.pfx** in the **%BTSSolutionsPath%\B2B\TPM\Certificate** folder. Accept all of the default values including **Password**.

In the **Certificates** snap-in, expand **Certificates - Current User**, right-click **Trusted Root Certificate Authorities**, point to **All Tasks**, and then click **Import**.

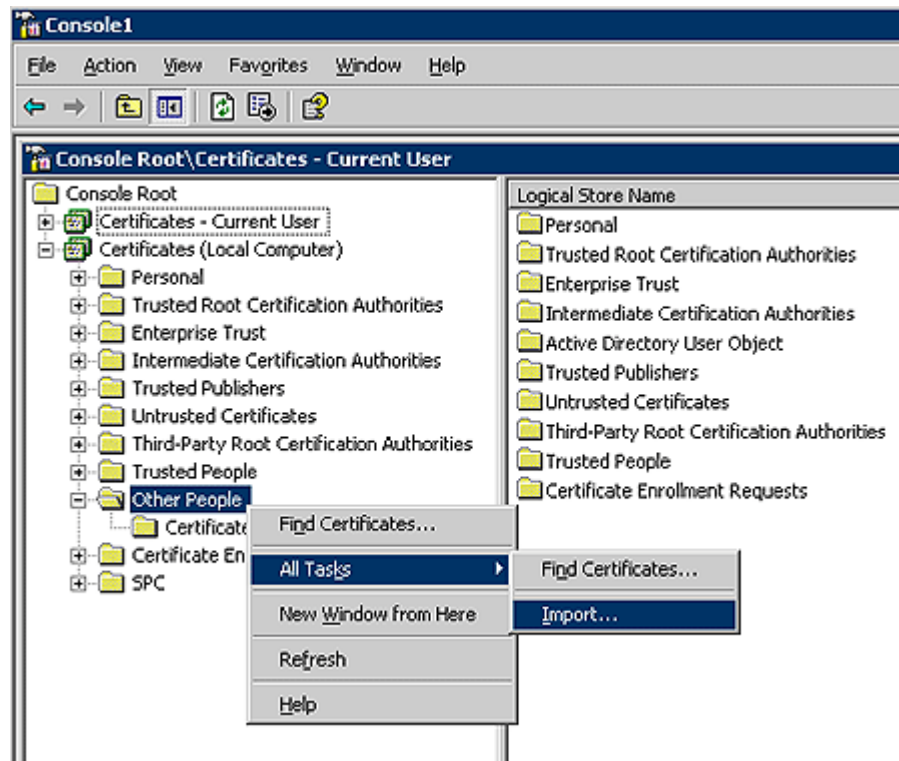


Using the **Certificate Import Wizard**, import the myserver.pfx in the %BTSSolutionsPath%\B2B\TPM\Certificate folder. Accept all of the default values including **Password**.

In the Microsoft Management Console, add the **Certificates** snap-in for **Computer account** of **Local computer**.



In the **Certificates** snap-in, expand **Certificates (Local Computer)**, right-click **Other People**, point to **All Tasks**, and then click **Import**.



Using the **Certificate Import Wizard**, import the myserver.pfx in the %BTSSolutionsPath%\B2B\TPM\Certificate folder. Accept all of the default values including **Password**.

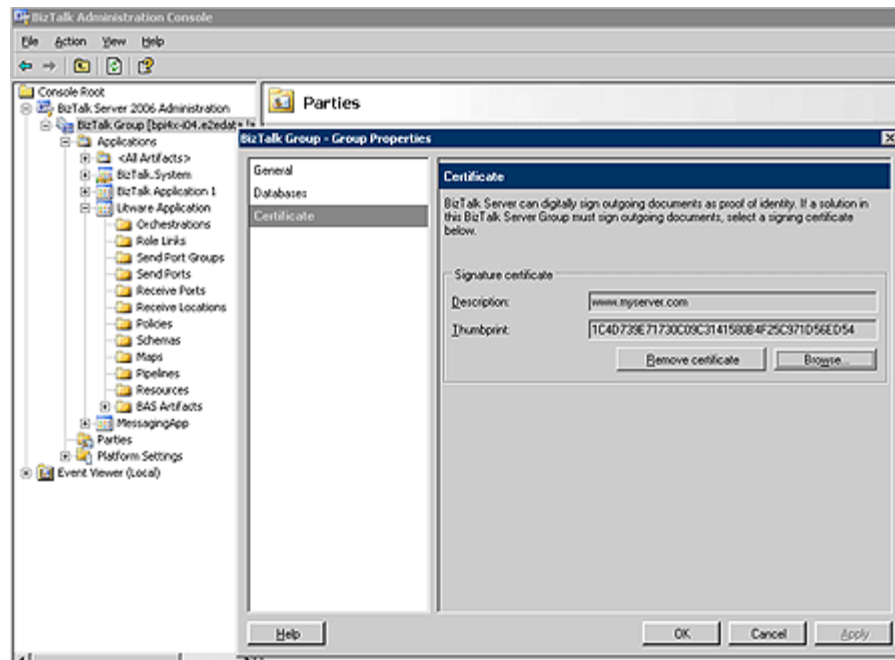
Click **Start**, point to **All Programs**, point to **Microsoft BizTalk Server 2006**, and then click **BizTalk Server Administration**.

Right-click **BizTalk Group**, and then click **Properties**.

On the **Group Properties** dialog box, click **Certificate**, click **Browse**.

On the **Select Certificate** dialog box, select the certificate that you installed, and then close all of the dialog boxes.





## Compile the Business-to-Business Solution

### To compile the Business-to-Business Solution

Open a command prompt, change the directory to the %BTSSolutonsPath%\B2B folder, type **Setup.bat**, and then press ENTER. The Setup.bat performs the following tasks:

- Creates a unique strong name key (SNK) for signing the assemblies of the B2B solution.

- Extracts the public key token from the SNK.

- Updates the binding files with the new public key token.

- Compiles the Microsoft.Samples.BizTalk.Litware solution.

- Compiles the SSOApplicationConfig solution in the %BTSSolutonsPath%\Common\SSOApplicationConfig folder.

## Register BizTalk Server on the BAS site

### To register BizTalk Server on the BAS site

Click **Start**, point to **All Programs**, point to **Microsoft BizTalk Server 2006**, and then click **Business Activity Services Site**.

- On the **Home** page, click **BizTalk Servers**.

- On the **BizTalk Servers** page, click **Register BizTalk Server**.

On the **BizTalk Servers: New Item** page, enter the registration information as the following table, and then click **Refresh Host Lists**.

Name	Value
Registration name	<Any value>
BizTalk Management Database Server Name	localhost
BizTalk Management database	BizTalkMgmtDb

On the **BizTalk Servers: New Item** page, select **BizTalkServerApplication** in the **Outbox Receive Location Host** and **Parameter Services Host** boxes, and then click **Save and Close**.

## Configure the hosts as Authentication Trusted

### To configure the hosts as Authentication Trusted

In the **BizTalk Server Administration** console, expand **BizTalk Group**, expand **Platform Settings**, and then expand **Host Instances**.

Right-click **BizTalkServerApplication**, and then click **Stop**.

Right-click **BizTalkServerApplication** again, and then click **Delete**.

Expand **Hosts**, right-click **BizTalkServerApplication**, and then click **Properties**. On the **Host Properties** page, select **Authentication Trusted**, click **OK** to close the dialog box.

Right-click **BizTalkServerIsolatedHost**, and then click **Properties**. On the **Properties** page, select **Authentication Trusted**, click **OK** to close the dialog box.

Right-click **Host Instances**, point to **New**, and then click **Host Instance**. On the **Host Instances Properties** page, select **BizTalkServerApplication** in the **Host name** box, click **Configure**, enter user information for the BizTalk service, and then close all of the dialog boxes.

## Install the Business-to-Business Solution

### To install the Business-to-Business Solution

Click **Start**, point to **All Programs**, point to **Microsoft Visual Studio 2005**, point to **Visual Studio Tools**, and then click **Visual Studio 2005 Command Prompt**.

At the **Visual Studio 2005 Command Prompt**, change the directory to the %BTSSolutionsPath%\B2B\Utilities\bin\debug folder, and then run the following command to add Utilities component to the Global Assembly Cache (GAC).

```
gacutil.exe /i Microsoft.Samples.BizTalk.Litware.Utilities.dll
```

At the **Visual Studio 2005 Command Prompt**, change the directory to the %BTSSolutionsPath%\B2B\PartyResolutionComponent\bin\debug folder, and then run the following command to add PartyResolutionComponent to the GAC.

```
gacutil.exe /i  
Microsoft.Samples.BizTalk.Litware.PartyResolutionComponent.dll
```

At a command prompt, change the directory to the %BTSSolutionsPath%\B2B\Scripts folder, type Locations.cmd, and then press ENTER. The command file creates the folders in the C drive for the adapters that the B2B solution will use.

In Windows Explorer, browse to the C:\Litware folder, and then make sure that the BizTalk service account has read/write permissions to the following folders:

BrokerNew

BrokerUpdate

EDIIn

EDIOut

GetOrder

GetProfile

UpdateUserBASRL

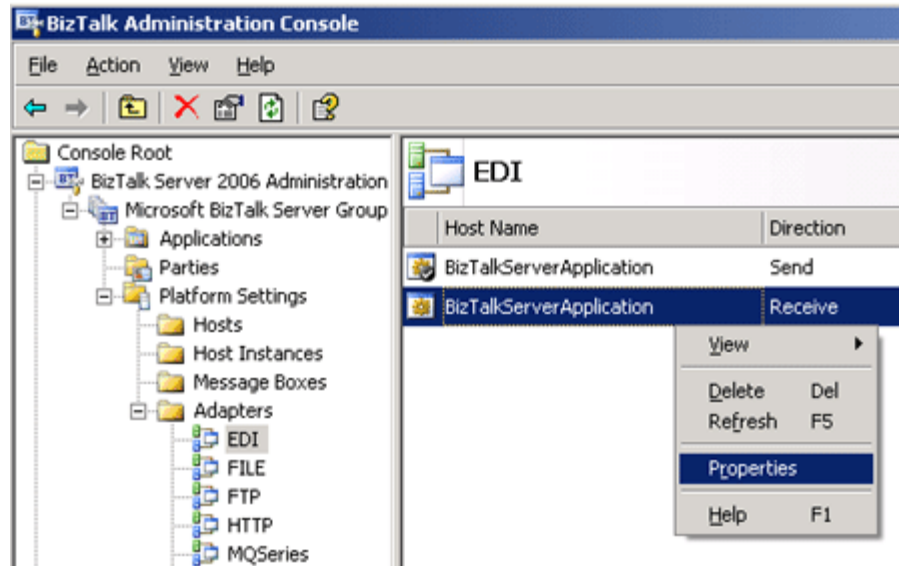
Vendor1

Vendor2

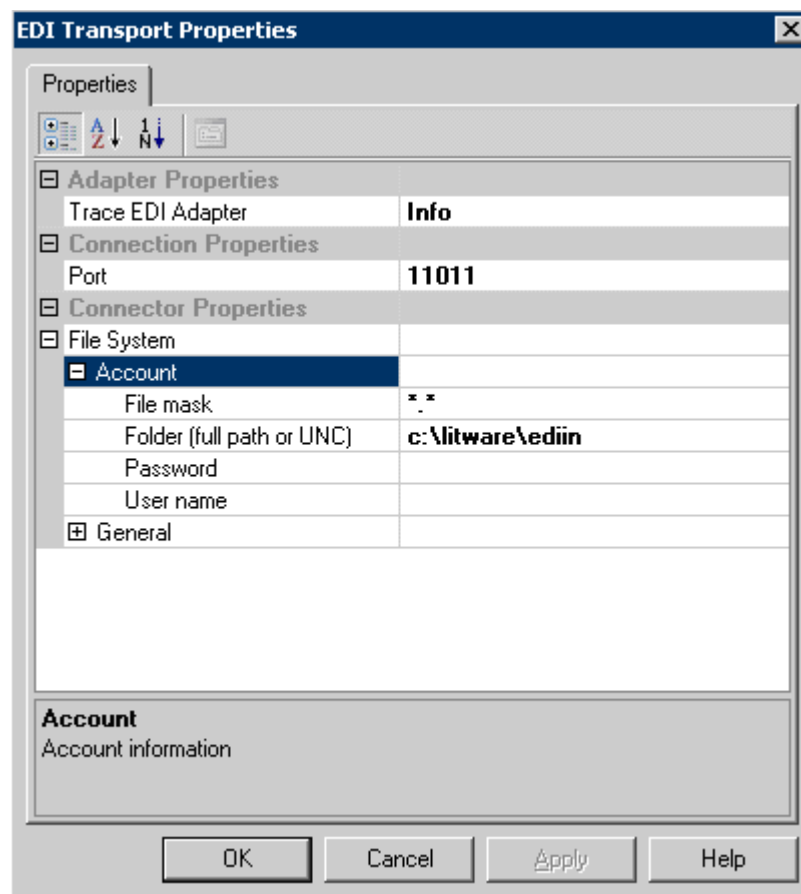
Click **Start**, point to **All Programs**, point to **Administrative Tools**, and then click **Services**. Using the **Services** console, make sure that the **BizTalk Base EDI service** is running.

In the **BizTalk Server Administration** console, expand **BizTalk Group**, expand **Platform Settings**, expand **Adapters**, and then click **EDI**.

Right-click **BizTalkServerApplication** with **Receive** direction, click **Properties**. On the **Adapter Handler Properties** dialog box, click **Properties**.



On the **Transport Properties** dialog box, expand **FileSystem**, expand **Account**, type **C:\Litware\EDI In** in the **Folder** text box, and then close all of the dialog boxes.



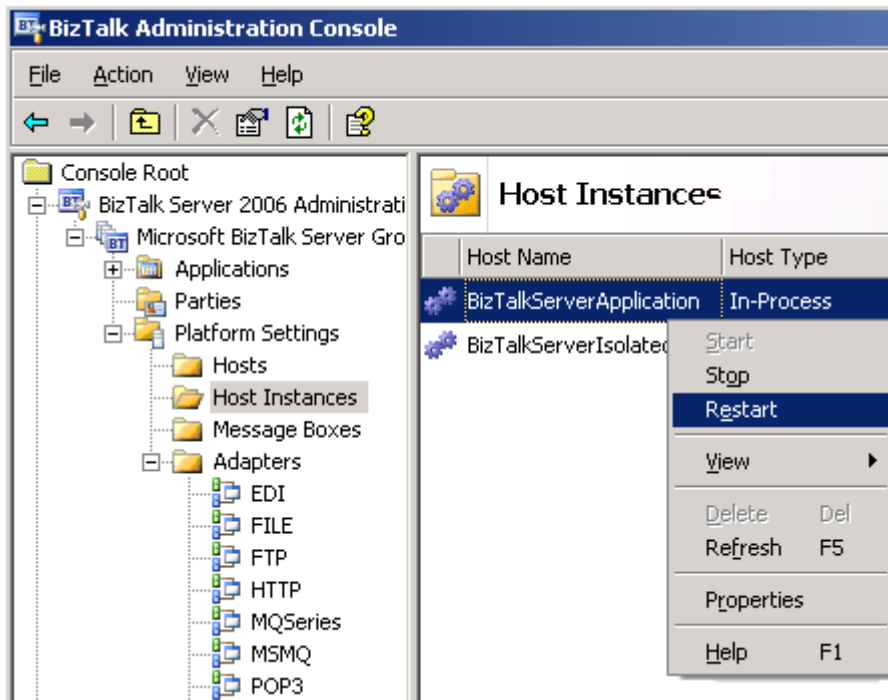
In the **BizTalk Server Administration** console, expand **BizTalk Group**, expand **Platform Settings**, expand **Adapters**, and then click **SMTP**.

Right-click **BizTalkServerApplication** with **Send** direction, click **Properties**. On the **Adapter Handler Properties** dialog box, click **Properties**.

On the **Transport Properties** dialog box, type **localhost** in the **SMTP server name** text box, and then type **administrator@<Your Domain Name>** in the **From** text box. This enables users to receive email notification when messages are processed.

Close all of the dialog boxes.

In the **BizTalk Server Administration** console, expand **BizTalk Group**, expand **Platform Settings**, expand **Host Instances**, right-click **BizTalkServerApplication**, and then click **Restart** to reflect new settings of the Base EDI Adapter.



In Windows Explorer, browse to the %BTSSolutionsPath%\B2B folder, and then open the Microsoft.Samples.BizTalk.Litware.sln with Microsoft Visual Studio 2005.

On the **Solution Explorer**, right-click Microsoft.Samples.BizTalk.Litware, and then click **Deploy Solution**.

In the **Internet Information Services (IIS) Manager**, expand **Web Sites**, right-click the **Default Web Site**, point to **New**, and then click **Virtual Directory** to run **Virtual Directory Creation Wizard**.

Using the **Virtual Directory Creation Wizard**, create the following virtual directory for the proxy Web service for the adapter version:

Alias = TPMAccessWS

PATH = <BizTalk Install Directory>\SDK\Scenarios\B2B\TPMAccessWS

Access Permissions = Read, Run scripts

Expand the **Default Web Site**, right-click TPMAccessWS, click Properties. On the **Properties** dialog box, click the **Virtual Directories** tab, and then in the **Application Pool** box, select the same application pool that is used by the TPPubWS Web application.

In the **Internet Information Services (IIS) Manager**, right-click **Default SMTP Virtual Server**, and then click **Properties**.

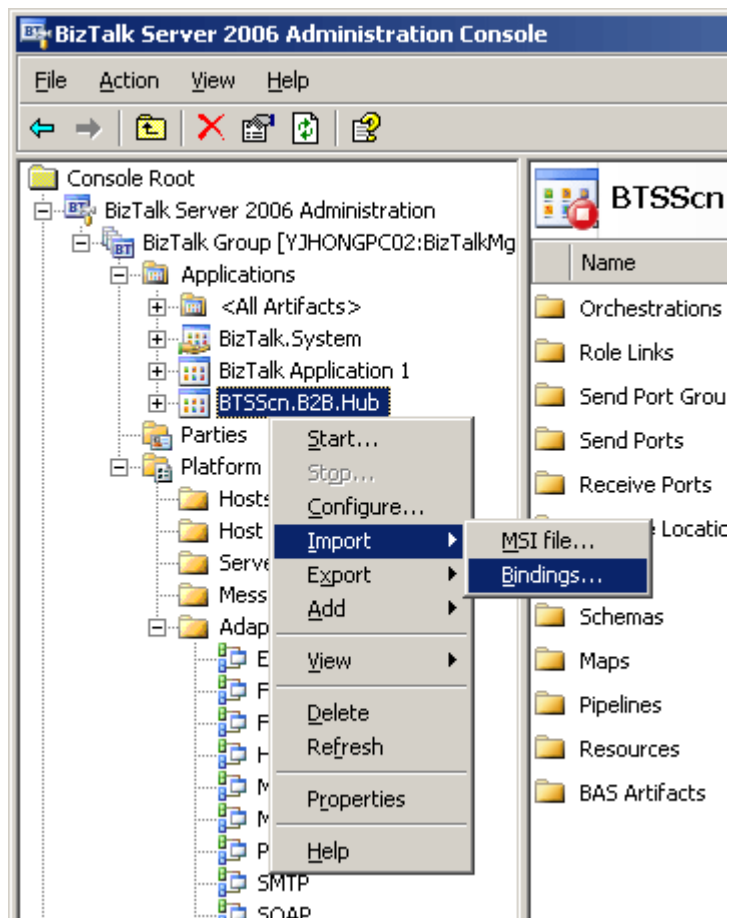
On the **Virtual Server Properties** page, click **Access**, and then click **Relay**.

On the **Relay Restriction** page, click **All except the list below**.

Close all of the dialog boxes.

In the **BizTalk Server Administration** console, expand **BizTalk Group**, and then expand **Applications**.

Right-click BTSScn.B2B.Hub application, point to **Import**, click **Bindings**, select the PartyPorts.xml in the %BTSolutionsPath%\B2B\Bindings folder, and then click **OK**.



Click **Start**, point to **All Programs**, point to **Microsoft SQL Server**, and then click **Query Analyzer**.

On the **Connect to SQL Server** dialog box, type <**Your Computer Name**> in the **SQL Server** text box, select **Windows authentication**, and then click **OK**.

Open the Order.SQL in the %BTSSolutionsPath%\B2B\OrderSystem\SQL folder, and then run the SQL query. It creates a database to log the incoming orders.

Open a command prompt, and then change the directory to the %BTSSolutionsPath%\B2B\Scripts folder.

At the command prompt, open the LitWareSSOConfiguration.xml file using Notepad, and then review the contents of the file. This file defines the configuration store application in SSO that the scenario uses to keep configuration parameters. No changes to this file are necessary.

At the command prompt, open the CreateLitwareApplication.cmd file using Notepad, review and change the contents of the file. This command file saves the connection string for the OrderSystemDB into the SSO affiliated application. No changes to this file are necessary.

At the command prompt, type **CreateLitwareApplication.cmd**, and then press ENTER.

## **Create the profile using the BAS site**

### **To create the profile using the BAS site**

In Windows Explorer, browse to the %BTSSolutionsPath%\B2B\TPM\PartnerData folder, and then open MyProfile.xml with Microsoft InfoPath 2003.

On the toolbar, click **Contacts**, and then type <**Your Email Address**> in the **E-mail address** text box.

**Microsoft Office InfoPath 2003**

File Edit View Insert Format Tools Table Help

General Information Addresses **Contacts** Print View Submit Profile Help

**My Profile** Microsoft BizTalk Server

**Contacts**  
Use the Contacts tab to enter contact information that you want to share with a partner. You can enter multiple sets of contact information in each profile. For example, you could add both a shipping contact and a billing contact.

Contact name: MyContact

Contact description: MyContact

User Name: MyContact

Password: [Password Field]

Contact type: MyContactType

Telephone number: 999-999-9999 Telephone description: 999-999-9999

☒ Insert telephone

Fax number: 999-999-9999 Fax description: 999-999-9999

☒ Insert Fax

E-mail address: someone@microsoft.com E-mail description: someone@microsoft.com

☒ Insert e-mail

☒ Insert contact

## Create the trading partner using the BAS site

### To create the trading partner using the BAS site

In Windows Explorer, browse to the %BTSSolutionsPath%\B2B\TPM\PartnerData folder, and then open TestPartnerExtended.xml with Microsoft InfoPath 2003.

On the toolbar, click **Contacts**, and then type <Your Email Address> in the **E-mail address** text box.

On the toolbar, click **Submit Profile**.

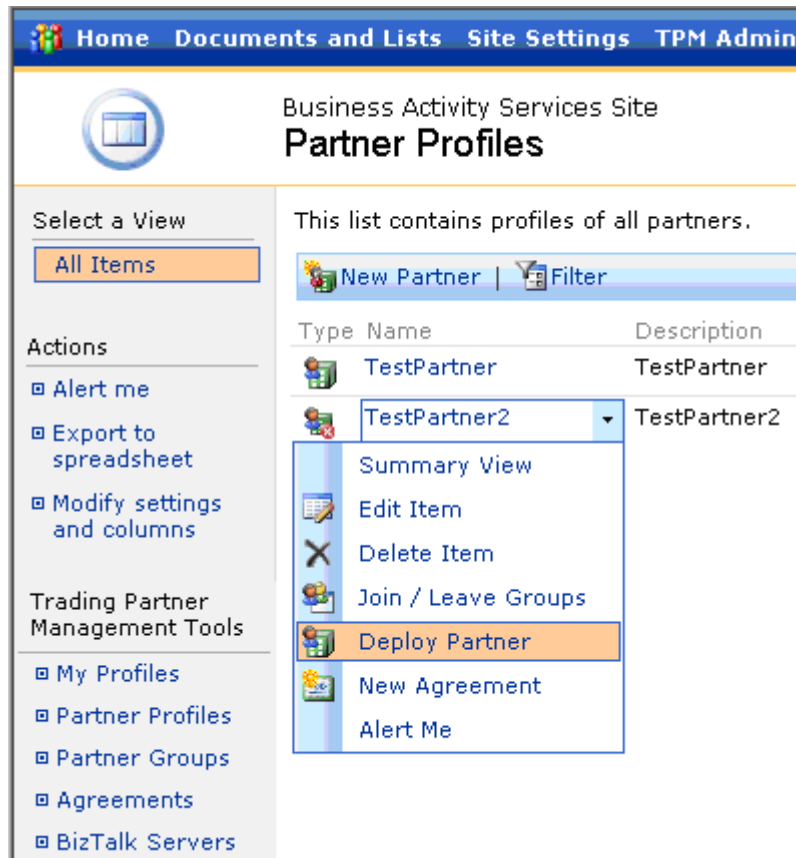
.Click **Start**, point to **All Programs**, point to **Microsoft BizTalk Server 2006**, and then click **Business Activity Services Site**.

On the **Home** page, click **Partner Profiles**.

On the **Partner Profiles** page, click TestPartner that you submitted with the TestPartnerExtended.xml, and then click **Deploy Partner**.

On the **Deploy Partner** page, select the registration name that you created in the step "To register BizTalk Server on the BAS site", and then click **Deploy**.





Repeat the same steps as the TestPartnerExtended.xml for submitting and deploying the TestPartner2Extended.xml the %BTSSolutionsPath%\B2B\TPM\PartnerData folder.

## Deploy the Business-to-Business Solution

### To deploy the Business-to-Business Solution

In the **BizTalk Server Administration** console, expand **BizTalk Group**, and then expand **Applications**.

Right-click BTSScn.B2B.Hub application, point to **Import**, click **Bindings**, select the OtherPorts.xml in the %BTSSolutionsPath%\B2B\Bindings folder, and then click **OK**.

Right-click BTSScn.B2B.Hub application, point to **Import**, click **Bindings**, select the BindingOrderSystem.xml in the %BTSSolutionsPath%\B2B\Bindings folder, and then click **OK**.

Right-click BTSScn.B2B.Hub application, point to **Import**, click **Bindings**, select the BindingOrchestration.xml in the %BTSSolutionsPath%\B2B\Bindings folder, and then click **OK**.

In the **Internet Information Services (IIS) Manager**, expand the **Web Sites**, expand the **Default Web Site**, right-click HWSMessages, and then click **Properties**.

On the **Properties** page, click **Directory Security** tab, and then click **Edit** to modify **Authentication and access control**. Select **Enable anonymous access**. Close all of the dialog boxes.

At a command prompt, perform an IISRESET.

In the **BizTalk Server Administration** console, expand **BizTalk Group**, expand **Platform Settings**, expand **Host Instances**, right-click **BizTalkServerApplication**, and then click **Restart** to reflect new settings.

In the **BizTalk Server Administration** console, expand **BizTalk Group**, expand **Applications**, expand **Send Ports**, right-click **SendToOrderSystem\_HTTP**, click **Properties**, and then click **Configure** on the Properties dialog box.

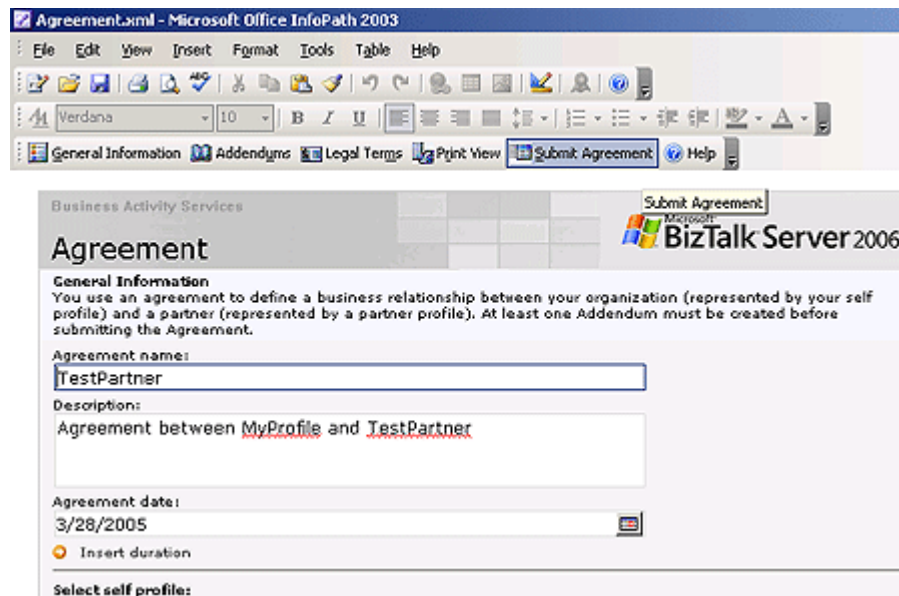
On the **Transport Properties** dialog box, click **General** tab, and then make sure that **Enable chunk encoding** is not selected.

On the **Transport Properties** dialog box, click **Authentication** tab, and then make sure that **Kerberos** is selected in the **Authentication type** box.

Close all of the dialog boxes.

In Windows Explorer, browse to the %BTSSolutionsPath%\B2B\TPM\PartnerData folder, and then open **Agreement.xml** with Microsoft InfoPath 2003.

On the toolbar, click **Submit Agreement**. It creates new agreement between the **TestPartner** and **MyProfile**.



The screenshot shows the Microsoft Office InfoPath 2003 application window titled "Agreement.xml - Microsoft Office InfoPath 2003". The menu bar includes File, Edit, View, Insert, Format, Tools, Table, and Help. The toolbar contains various icons for file operations and formatting. Below the toolbar, there are tabs for General Information, Addendums, Legal Terms, Print View, Submit Agreement, and Help. The main content area displays the "Agreement" form, which is part of the "Business Activity Services" site. The form includes a "General Information" section with instructions: "You use an agreement to define a business relationship between your organization (represented by your self profile) and a partner (represented by a partner profile). At least one Addendum must be created before submitting the Agreement." The form fields are: "Agreement name:" with the value "TestPartner", "Description:" with the value "Agreement between MyProfile and TestPartner", and "Agreement date:" with the value "3/28/2005". There is also an "Insert duration" button and a "Select self profile:" section.

Click **Start**, point to **All Programs**, point to **Microsoft BizTalk Server 2006**, and then click **Business Activity Services Site**.

On the **Home** page, click **Agreements**.

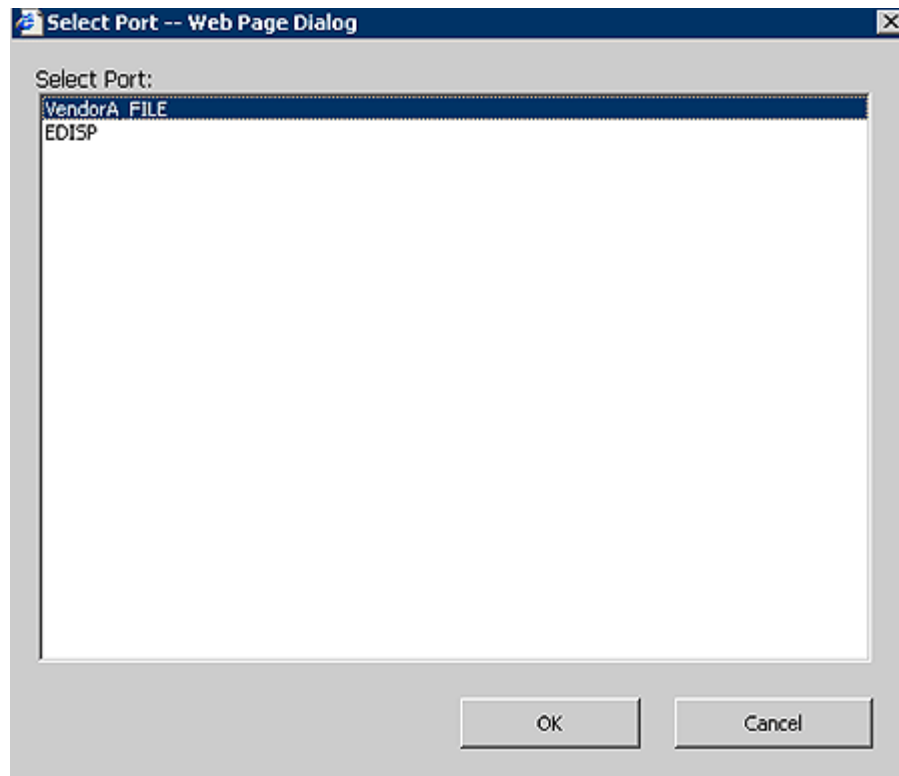
On the **Agreements** page, click **Activate Agreement** from the drop-down list for the TestPartner. It opens the agreement with Microsoft InfoPath 2003.

The screenshot shows the 'Business Activity Services Site Agreements' page. The top navigation bar includes 'Home', 'Documents and Lists', 'Site Settings', and 'TPM Admin'. The page title is 'Business Activity Services Site Agreements'. On the left, there is a sidebar with 'Select a View' (All Items), 'Actions' (Alert me, Export to spreadsheet, Modify settings and columns), and 'Trading Partner Management Tools' (My Profiles, Partner Profiles, Partner Groups, Agreements, BizTalk Servers). The main content area shows a list of agreements with the following table:

Type	Name	Description
Agreement	TestPartner	Agreement
Agreement	Summary View	Agreement

A context menu is open for the 'TestPartner' agreement, showing options: 'Summary View', 'Edit Item', 'Delete Item', 'Activate Agreement' (highlighted), and 'Alert Me'.

On the **Activate Agreement** page in Microsoft InfoPath 2003, select VendorA\_FILE for all seven **Operation Ports** boxes, and then click **Submit Mapping**.



In Windows Explorer, browse to the %BTSSolutionsPath%\B2B\TPM\PartnerData folder, and then open Agreement2.xml with Microsoft InfoPath 2003.

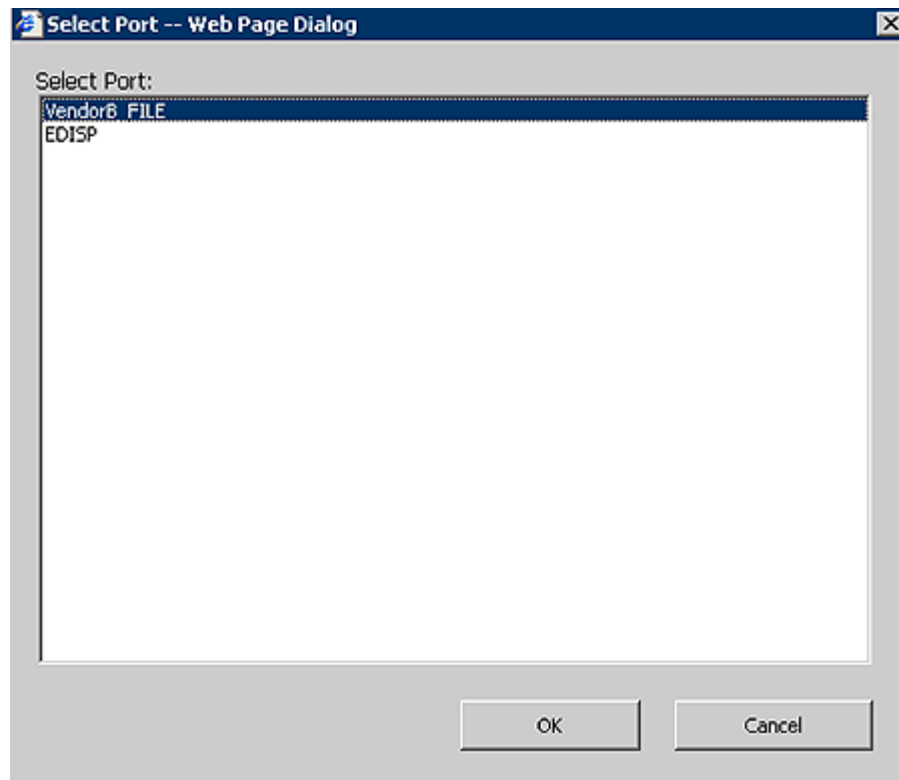
On the toolbar, click **Submit Agreement**. It creates new agreement between the TestPartner2 and MyProfile.

Click **Start**, point to **All Programs**, point to **Microsoft BizTalk Server 2006**, and then click **Business Activity Services Site**.

On the **Home** page, click **Agreements**.

On the **Agreements** page, click **Activate Agreement** from the drop-down list for the TestPartner2. It opens the agreement with Microsoft InfoPath 2003.

On the **Activate Agreement** page in Microsoft InfoPath 2003, select VendorB\_FILE for all seven **Operation Ports** boxes, and then click **Submit Mapping**.



Run the Business-to-Business Solution. For more information about running the solution, see [How to Install the Business-to-Business Solution](#).

## How to Run the Business-to-Business Solution

The following steps describe how to run and validate the Business-to-Business solution on a single computer.

Run and validate the Business-to-Business Solution

### Run and validate the Business-to-Business Solution

#### To start the BTSScn.B2B.Hub application

In the **BizTalk Server Administration** console, expand **BizTalk Group**, expand **Platform Settings**, expand **Host Instances**, right-click **BizTalkServerApplication**, and then click **Restart**.

In the **BizTalk Server Administration** console, expand **BizTalk Group**, and then expand **Applications**, right-click **BTSScn.B2B.Hub**, click **Start**, and then click **Start** on the **Start Application** dialog box.

#### To run and validate the Business-to-Business Solution

Create new orders:

In Windows Explorer, browse to the %BTSSolutoinsPath%\B2B\SampleMessages folder, and then copy the Basket.xml and past it to the C:\Litware\BrokerNew folder.

In Windows Explorer, browse to the C:\Litware\Vendor1 folder, and then check new messages coming in for validation.

Check an email notification that is sent to the email address entered when creating a trading partner using the BAS site) for validation.

#### Update existing orders:

At a command prompt, change the directory to the %BTSSolutoinsPath%\B2B\SampleMessages, type the following command, and then press ENTER.

**copy PO.xml POUpdate.xml**

Open POUpdate.xml using Notepad, and then modify the timestamp in the **LastModified** attribute of **Basket** node in the file. Make sure that the **SoldToID** and **OrderGroupID** attributes are same as the existing order you created in the previous step.

Create a signed message out of the POUpdate.xml. For this walkthrough use POUpdateSigned.xml for the name of the singed message. For more information on how to sign messages, see Certificates and Signing Messages.

In Windows Explorer, copy the POUpdateSigned.xml and past it to the C:\Litware\BrokerUpdate folder.

In Windows Explorer, browse to the C:\Litware\Vendor1 folder, and then check new messages coming in for validation.

Check an email notification that is sent to the email address entered when creating a trading partner using the BAS site) for validation

#### Retrieve the information about existing orders:

Create a signed message out of the OrderInquery.xml in the %BTSSolutoinsPath%\B2B\SampleMessages folder. For this walkthrough, use GetOrderSigned.xml for the name of the signed message.

In Windows Explorer, copy the GetOrderSigned.xml and past it to the C:\Litware\GetOrder folder.

In Windows Explorer, browse to the C:\Litware\Vendor1 folder, and then check new messages coming in for validation.

#### Retrieve partner profiles:

Create a signed message out of the RequestProfile.xml in the %BTSSolutoinsPath%\B2B\SampleMessages folder. For this walkthrough, use RequestProfileSigned.xml for the name of the signed message.

In Windows Explorer, copy the RequestProfileSigned.xml and past it to the C:\Litware\GetProfile folder.

In Windows Explorer, browse to the C:\Litware\Vendor1 folder, and then check new messages coming in for validation.

Update partner profiles:

In Windows Explorer, copy the result message from the GetProfileSigned.xml, and past it to the C:\Litware\UpdateUserBASRL folder.

In Windows Explorer, browse to the C:\Litware\Vendor1 folder, and then check new messages coming in for validation.

Create new orders using EDI messages:

In Windows Explorer, browse to the <BizTalk Install Directory>\EDI\Adapter\Getting Started with EDI\Contoso\Pickup folder, and then copy ContosoPickupInstance.edi file to the C:\Litware\EDIIn folder.

In Windows Explorer, browse to the C:\Litware\Vendor1 folder, and then check new messages coming in for validation.

In Windows Explorer, browse to the C:\Litware\Vendor2 folder, and then check new messages coming in for validation.

## **Prerequisites of the B2B Solution Production Servers**

The Business-to-Business Solution can consist of different computers, and each computer plays an integral part in the overall sample. This section describes the hardware and software prerequisites for each computer.

Domain Controller

Windows Server 2003 Enterprise Edition

Two BizTalk Servers

Windows Server 2003 Enterprise Edition

BizTalk Server 2006

Two Network Interface Cards (NIC) per server for Network Load Balancing (NLB)

Two SQL Servers for BizTalk Servers

Windows Server 2003 Enterprise Edition

SQL Server 2000 and SP4

Two NICs per server for SQL Cluster

A shared storage resource

Administration Client for BizTalk Servers

Windows Server 2003 Enterprise Edition

BizTalk Server 2006

Web Server for Business Activity Service (BAS)

Windows Server 2003 Enterprise Edition

.NET Framework 2.0

## How to Prepare the B2B Solution for Deployment

The following steps describe how to prepare the resources for a business-to-business solution for deployment to the production servers.

To prepare resources to be deployed to BizTalk Servers

To prepare resources to be used from Administration Client

To prepare resources to be deployed to Trading Partner Management (TPM) server

### Prerequisites

Before deploying the business-to-business Solution to the production servers, you must finish the steps shown in the topic Developer Machine Setup for the Business-to-Business Solution.

### To prepare resources for deployment to BizTalk Servers

Compile the solution using Release Configuration

Search and terminate all the service instances of the business-to-business solution.

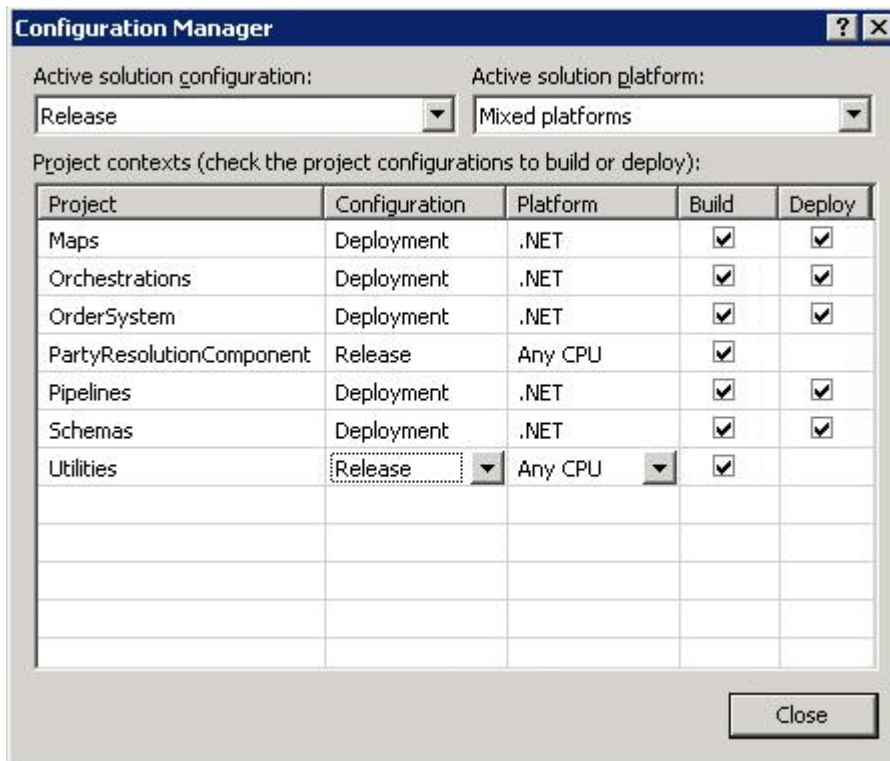
Stop the host instance that is running the business-to-business solution. For more information about how to stop the host instance, see How to Stop a Host Instance.

Click **Start**, point to **All Programs**, point to **Microsoft Visual Studio 2005**, and then click **Microsoft Visual Studio 2005**.

On the **File** menu, point to **Open**, and then click **Project/Solution**. In the **Open Project** dialog box, navigate to the %BTSSolutionsPath%\B2B\ folder, and then open Microsoft.Samples.BizTalk.Litware.sln.



On the **Build** menu, click **Configuration Manager**, configure it as the following image, and then exit the dialog box.



On the **File** menu, click **Save All**, and then exit Visual Studio 2005.

To open the Visual Studio 2005 command prompt, click **Start**, point to **All Programs**, point to **Microsoft Visual Studio 2005**, point to **Visual Studio Tools**, and then click **Visual Studio 2005 Command Prompt**.

At the **Visual Studio 2005 Command Prompt**, run the following command line to set the %BTSSolutionsPath% environment variables to indicate the base folder of the E2E solutions.

```
set      BTSSolutionsPath=%ProgramFiles%\Microsoft      BizTalk      Server
2006\SDK\Scenarios
```

### To prepare resources for use from the Administration Client

Compile the business-to-business solution with Release Configuration.

In the folder c:\deploy, create the folder AdminClient.

In TPM\PartnerData folder, copy all of the files to the AdminClient folder.

In the TPM\SQL folder, copy the file Contact.SQL to the AdminClient folder.

To copy the certificate for signed messages, copy the file TPM\Certificate\myserver.pfx to the BizTalkServer folder.

In the TPM\InfoPathform folder, copy all files to the BizTalkServer folder.

In the OrderSystem\SQL folder, copy the file Copy Order.SQL to the AdminClient folder.

Copy the scrip files and utility application for SSO configuration.

In the Scenarios\B2B\Scripts folder, copy the files CreateLitwareApplication.cmd and LitwareSSOConfiguration.xml to the AdminClient folder.

Copy BTSScnSSOApplicationConfig.exe in the ..\Common\SSOApplicationConfig\bin folder.

Create the binding file for BTSScn.B2B.Hub application.

Navigate to **Console Root->BizTalk Server 2006 Administration->BizTalk Group->Applications**.

Right-click BTSScn.B2B.Hub, point to **Export**, and then click **Bindings**.

From the current application, select **Export all bindings**.

Type c:\deploy\AdminClient\BTSScn.B2B.Hub.BindingInfo.xml in the **Export to file** box, and then click **OK**.

Copy the file c:\deploy\BizTalkServer\BTSScn.B2B.Hub.msi to the AdminClient folder.

### **To prepare resources for deployment to the Trading Partner Management (TPM) server**

In the c:\deploy folder, create the folder TPMServer.

Copy the TPMAccessWS folder in the c:\deploy.

In the TPM\Schemas folder, copy all the files to the TPMServer folder you created.

In the TPM\TPMSchema folder, copy the file TPMUserSchemaNotEncrypted.XML to the TPMServer folder.

To copy the certificate for signed messages, copy the file TPM\Certificate\myserver.pfx to the BizTalkServer folder.

### **To prepare resources for deployment to the legacy system**

In the folder c:\deploy, create the folder LegacyServer.

In the folder OrderSystem\SQL, copy the file Order.SQL to the LegacyServer folder.

## Preparing the Production Servers for the B2B Solution

The following steps describe how to prepare the production servers for installing the BPM solution.

To create a domain controller

To create global domain security groups and user accounts that the B2B Solution will use.

To configure SQL Cluster

To install the first BizTalk Server

To configure the first BizTalk Server

To install the second BizTalk Server

To configure the second BizTalk Server

To setup TCP/IP for configuring Network Load Balancing (NLB) between BizTalk Servers

To install Administration Client

To configure the Administration Client

To configure SQL Server in order to enable Windows SharePoint Services to access the required databases

To install Trading Partner Management (TPM) server

To configure Trading Partner Management (TPM) server

To install legacy system

The following table summarizes the server name and the applications to be deployed.

Network name	Applications/Services	Description
E2E-BTS1	Master Secret Service	BizTalk Server
	HTTP Adapter	
	File Adapter	
	SMTP Adapter	
	EDI Adapter	
E2E-BTS2	HTTP Adapter	BizTalk Server

	File Adapter SMTP Adapter EDI Adapter	
E2E-SQL1	BizTalk databases SSO database	SQL Server
E2E-SQL2	BizTalk databases SSO database	SQL Server
E2E-ADMIN	BizTalk Administration tools	Administration Client
E2E-WEB	SharePoint Adapter BAS	Trading Partner Management (TPM) server
E2E-LEGACY	OrderSystemDB	The server simulating legacy systems
E2E-DC	Domain controller	Domain controller

## Prerequisites

Before deploying the B2B Solution to the production servers, you have to finish Developer Machine Setup for the Business-to-Business Solution.

### To create a domain controller

The B2B solution assumes that this server is named E2E-DC.

Run **dcpromo.exe** to create a new domain controller, new domain, and new forest.

The full DNS name of the new domain is **contoso.com**.

The domain NetBIOS is **CONTOSO**.

### To create global domain security groups and user accounts that the B2B solution will use.

Create the following new organizational unit using dsadd.exe command line.

BTSAccounts

BTSAccounts\btsntsvc

BTSAccounts\Group

BTSAccounts\Groups\Global

BTSAccounts\Users

Create the following new domain groups in BTSAccounts\Groups\Global organizational unit using dsadd.exe command line.

SSO Administrators

SSO Affiliate Administrators

BizTalk Server Administrators

BizTalk Server Operators

BAS Enabled Hosts

BizTalk BAS Administrators

BizTalk BAS Managers

BizTalk BAS Users

BizTalk BAS Web Services Group

EDI Subsystem Users

SharePoint Enabled Hosts

BizTalk Tracking Host Users

BizTalk B2B Hub Host Users

BizTalk EDI Adapter Host Users

BizTalk File Adapter Trusted Host Users

BizTalk HTTP Adapter Isolated Host Users

BizTalk HTTP Adapter Trusted Host Users

BizTalk SharePoint Adapter Host Users

BizTalk SMTP Adapter Host Users

BizTalk SOAP Adapter Trusted Host Users

## SQL Service Group

Create domain users in the BTSAccounts\Users organizational unit, and then add them to the groups you created. You can use dsadd.exe command line.

Domain user name	Domain name description	Members of
ssoadmin	SSO Administrator	SSO Administrators
btsadmin	BizTalk Administrator	SSO Affiliate Administrators BizTalk Server Administrators SSO Administrators
btsinstall	BizTalk Installation	SSO Administrators
ssoslave1	SSO Service	SSO Administrators
ssomaster	SSO Master Secret	SSO Administrators
sqlservice	SQL Service	SQL Service Group
btsoperator	BizTalk Server Operators	BizTalk Server Operators
basadmin		
basapppool		BizTalk BAS Web Services Group
basmgmtwebsvc		BizTalk BAS Web Services Group BizTalk Server Administrators
basplwebsvc		BizTalk BAS Web Services Group BizTalk BAS Users BAS Enabled Hosts
wsssvc		
wssadmin		
btsedisubsystemsvc	BizTalk Base EDI service	BizTalk EDI Adapter Host Users EDI Subsystem Users

		BizTalk Server Administrators
--	--	-------------------------------

Create domain users in the BTSAccounts\btsntsvc organizational unit, and then add them to the groups you just created. Check User cannot change password checkbox and uncheck Password never expires and User must change password at next logon checkboxes for the following accounts.

Domain user name	Domain name description	Members of
btstracking	BizTalk Tracking	BizTalk Tracking Host Users
btsb2bhub	BizTalk B2B Hub Orchestrations	BizTalk B2B Hub Host Users BizTalk Server Administrators
btsediadapter	BizTalk EDI Adapter	BizTalk EDI Adapter Host Users EDI Subsystem Users
btsfiletrusted	BizTalk File Adapter	BizTalk File Adapter Trusted Host Users SSO Administrators
btshttpisolated	BizTalk HTTP Receive Adapter	BizTalk HTTP Adapter Isolated Host Users
btshttptrusted	BizTalk HTTP Send Adapter	BizTalk HTTP Adapter Trusted Host Users SSO Administrators
btssharepointadapter	BizTalk SharePoint Adapter	BizTalk SharePoint Adapter Host Users SharePoint Enabled Hosts
btssmtpadapter	BizTalk SMTP Adapter	BizTalk SMTP Adapter Host Users
btssoaptrusted	BizTalk SOAP Send Adapter	BizTalk SOAP Adapter Trusted Host Users BizTalk SSO Administrators

### To configure SQL Cluster

Make sure to join all cluster nodes to contoso.com domain.

Add CONTOSO\btsinstall and CONTOSO\btsadmin to the local administrators group of E2E-SQL1 and E2E-SQL2.

Install and configure Windows Server 2003 Clustering Services.

Configure MS DTC.

Make sure to enable Network DTC Access.

Install and Configure SQL Cluster.

The B2B solution assumes that the SQL server name is E2E-SQL.

Use CONTOSO\sqlservice domain user account for SQL Server and SQL Server Agent service.

Use Windows Authentication Mode.

Make sure to configure SQL Server in Auto-start service

Install SQL Server 2000 SP4.

#### **To install the first BizTalk Server node.**

The B2B solution assumes that the first BizTalk Server name is E2E-BTS1.

Prepare the redistributable prerequisites CAB file.

Make sure to join the system to contoso.com domain.

Add CONTOSO\btsinstall and CONTOSO\btsadmin to the local Administrators group.

Logon using CONTOSO\btsinstall domain account.

Install IIS and enable Network DTC Access.

Navigate **Start->Control Panel->Add or Remove Programs**, click and **Add/ Remove Windows Components**.

Double click **Application Server** on Windows Component Wizard.

Install **Application Server Console, Enable network COM+ access, Enable network DTC access, Internet Information service (IIS)**.

Exit the all the dialog boxes.

Configure DTC

Open command line prompt.



Run the following command line to start Component Services Console.

**%SystemRoot%\system32\com\comexp.msc**

Navigate to **Console Root->Component Services->Computers->My Computer**. Then, Right click on **My Computer** and click **Properties**.

Click **MSDTC**, and click **Security Configuratoin....**

Make sure the following option boxes checked.

**Network DTC Access**

**Allow Remote Clients**

**Allow Remote Administrations**

**Allow Inbound**

**Allow Outbound**

**Enable XA Transactions**

Run SQL Server 2000 setup program.

Check **Client tools only** on **Installation Definition** dialog box in installing SQL Server 2000.

Install SQL Server 2000 SP4.

Run BizTalk Server 2006 setup program.

In Component Installation dialog box, select the following components to install:

**Server Runtime**

**Base EDI Adapter**

**Administration Tools**

**Enterprise Single Sign-On Administration**

**Enterprise Single Sign-On Master Secret Server**

In Redistributable Prerequisites dialog box, select "Automatically install the redistributable prerequisites from a CAB file" option, and choose the pre-downloaded CAB file in the step 1.

**To configure the first BizTalk Server node.**

Run Configuration.exe to start Microsoft BizTalk Server 2006 Configuration.

On the **Microsoft BizTalk Server 2006 Configuration page - Start** page,

Select **Custom configuration**.

Type E2E-SQL as **Database server name**.

Click **Configure**.

Click **Enterprise SSO** in the left pane, and in the right pane:

Select **Enable Enterprise Single Sign-On on this computer** checkbox.

Select **Create a new SSO system** option.

Choose CONTOSO\ssomaster as **Enterprise Single Sign-On Service** account.

Choose CONTOSO\SSO Administrators as **SSO Administrator(s)** role.

Choose CONTOSO\SSO Affiliate Administrators as **SSO Affiliate Administrator(s)** role.

Click **Enterprise SSO Secret Backup** in the left pane, and in the right pane:

Type Secret **backup password**, **Confirm password**, **Password reminder**.

Click **Group** in the left pane, and in the right pane:

Select **Enable BizTalk Server Group on this computer** checkbox.

Select **Create a new BizTalk Group** option.

Choose CONTOSO\BizTalk Server Administrators as **BizTalk Administrators Group**.

Choose CONTOSO\BizTalk Server Operators as **BizTalk Operators Group**.

Click **BizTalk Runtime** in the left pane, and in the right pane:

Select **Register the BizTalk Server runtime components**.

Select **Create BizTalk In-Process Host and Host Instance?** checkbox.

Type **EDIAdapter** in the **BizTalk Host Name** box.

Unselect **Create BizTalk Isolated Host and Host Instance?** checkbox

Type CONTOSO\btsediadapter for the **Account** of the **BizTalk Host Instance Account**.

Type CONTOSO\BizTalk EDI Adapter Host Users for the **Windows Group** of the **BizTalk Host Users Group**.

Click **EDI** in the left pane, and in the right pane,

Select **Enable Base EDI Adapter on this computer** checkbox.

Type **E2E-SQL** and **BizTalkEDIDb** for the **Server Name** and **Database Name** of the **BizTalk Base EDI database**.

Type **CONTOSO\btsedisubsystemsvc** for the **Account** of the **BizTalk Base EDI Service**.

Type **CONTOSO\EDI Subsystem Users** for the **Windows Group** of the **BizTalk Base EDI Users group**.

Click **Apply Configuration** in the tool bar.

Click **Configure** to start configuration. ☐ To install the second BizTalk Server node.

The B2B solution assumes that the second BizTalk Server name is E2E-BTS2.

Prepare the redistributable prerequisites CAB file.

Make sure to join the system to contoso.com domain.

Add CONTOSO\btsinstall and CONTOSO\btsadmin to the local Administrators group.

Logon using CONTOSO\btsinstall domain account.

Install IIS and enable Network DTC Access.

Navigate **Start->Control Panel->Add or Remove Programs**, click and **Add/ Remove Windows Components**.

Double click **Application Server** on Windows Component Wizard.

Install **Application Server Console**, **Enable network COM+ access**, **Enable network DTC access**, and **Internet Information service (IIS)**.

Exit the all the dialog boxes.

Configure DTC

Open command line prompt.

Run the following command line to start Component Services Console.

**%SystemRoot%\system32\com\comexp.msc**

Navigate to **Console Root->Component Services->Computers->My Computer**. Then, Right click on **My Computer** and click **Properties**.

Click **MSDTC**, and click **Security Configuratoin....**

Make sure the following option boxes checked.

**Network DTC Access**

**Allow Remote Clients**

**Allow Remote Administrations**

**Allow Inbound**

**Allow Outbound**

**Enable XA Transactions**

Run SQL Server 2000 setup program.

Check **Client tools only** on **Installation Definition** dialog box in installing SQL Server 2000.

Install SQL Server 2000 SP4.

Run BizTalk Server 2006 setup program.

In Component Installation dialog box, select the following components to install:

**Server Runtime**

**Base EDI Adapter**

In Redistributable Prerequisites dialog box, select "Automatically install the redistributable prerequisites from a CAB file" option, and choose the pre-downloaded CAB file in the step 1.

**To configure the second BizTalk Server node.**

Run Configuration.exe to start Microsoft BizTalk Server 2006 Configuration.

On the **Microsoft BizTalk Server 2006 Configuration page - Start** page,

Select **Custom configuration**.

Type E2E-SQL as **Database server name**.

Click **Configure**.

Click **Enterprise SSO** in the left pane, and in the right pane:

Select **Enable Enterprise Single Sign-On on this computer** checkbox.

Select **Join an existing SSO System** option.

Choose CONTOSO\ssoslave1 as **Enterprise Single Sign-On Service** account.

Click **Group** in the left pane, and in the right pane:

Select **Enable BizTalk Server Group on this computer** checkbox.

Select **Join an existing BizTalk Group** option.

Click **BizTalk Runtime** in the left pane, and in the right pane:

Select **Register the BizTalk Server runtime components**.

Unselect **Create BizTalk In-Process Host and Host Instance?** checkbox.

Unselect **Create BizTalk Isolated Host and Host Instance?** checkbox.

Click **EDI** in the left pane, and in the right pane:

Select **Enable Base EDI Adapter on this computer** checkbox.

Type **CONTOSO\btsedisubsystemsvc** for the **Account** of the **BizTalk Base EDI Service**.

Click **Apply Configuration** in the tool bar.

Click **Configure** to start configuration.

### **To setup TCP/IP for configuring Network Load Balancing (NLB) between the BizTalk Servers.**

Configure Network Load Balancing and TCP/IP components of the first BizTalk Server.

The BPM solution assumes that the Full Internet name under Cluster Parameters tab is E2E-BTS.contoso.com.

Configure Network Load Balancing and TCP/IP components of the second BizTalk Servers.

Add New Host for E2E-BTS in the E2E-DC.

### **To install the Administration Client**

The BPM solution assumes that the administration client is E2E-ADMIN.

Prepare the redistributable prerequisites CAB file.

Make sure to join the system to contoso.com domain.

Add CONTOSO\btsinstall and CONTOSO\btsadmin to the local Administrators group.

Logon using CONTOSO\btsinstall domain account.

Make sure to enable Network DTC Access and Network COM+ Access.

Navigate **Start->Control Panel->Add or Remove Programs**, click and **Add/ Remove Windows Components**.

Double click **Application Server** on Windows Component Wizard.

Make sure **Enable network DTC access** and **Enable network COM+ Access** checked.

Exit the all the dialog boxes.

Open command line prompt.

Run the following command line to start Component Services Console.

**%SystemRoot%\system32\com\comexp.msc**

Navigate to **Console Root->Component Services->Computers->My Computer**. Then, Right click on **My Computer** and click **Properties**.

Click **MSDTC**, and click **Security Configuratoin....**

Make sure the following check box enabled.

**Network DTC Access**

**Allow Remote Clients**

**Allow Remote Administrations**

**Allow Inbound**

**Allow Outbound**

**Enable XA Transactions**

Run BizTalk Server 2006 setup program.

In Component Installation dialog box, select the following components to install:

**Documentation**

**Administration Tools**

**Enterprise Single Sign-On Administration**

## Business Rule Component

In Redistributable Prerequisites dialog box, select "Automatically install the redistributable prerequisites from a CAB file" option, and choose the pre-downloaded CAB file in the step 1.

### To configure Administration Client.

Run Configuration.exe to start Microsoft BizTalk Server 2006 Configuration.

On the **Microsoft BizTalk Server 2006 Configuration page - Start** page,

Select **Custom configuration**.

Type E2E-SQL as **Database server name**.

Click **Configure**.

Click **Group** in the left pane, and in the right pane:

Select **Enable BizTalk Server Group on this computer** checkbox.

Select **Join an existing BizTalk Group** checkbox.

Type **E2E-SQL** and **BizTalkMgmtDb** for the **Server Name and Database Name** of the **BizTalk Management Database**.

Click **Apply Configuration** in the tool bar.

Click **Configure** to start configuration.

### To configure SQL Server in order to enable Windows SharePoint Services to access the required databases

Logon to E2E-SQL1 using CONTOSO\btsinstall account.

Click **Start**, point to **All Programs**, point to **Microsoft SQL Server**, and then click **Enterprise Manager**.

In the **Enterprise Manager**, expand **Microsoft SQL Servers**, expand **SQL Server Group**, expand E2E-SQL, expand **Security**, right-click **Logins**, and then click **New Login**.

In the **New Login** dialog box, type **CONTOSO\wss** in the **Name** box, click **Server Roles** tab, select **Security Administrators** and **Database Creators** check boxes, and then. Click OK.

Exit the Enterprise Manager.

### To install Trading Partner Management (TPM) server

The B2B solution assumes that the web server name is E2E-WEB.

Make sure to join the system to contoso.com domain.

Add CONTOSO\btsinstall and CONTOSO\btsadmin to the local Administrators group.

Logon using CONTOSO\btsinstall domain account.

Install IIS, ASP.NET and enable Network DTC Access.

Navigate **Start->Control Panel->Add or Remove Programs**, click and **Add/ Remove Windows Components**.

Double click **Application Server** on Windows Component Wizard.

Install **Application Server Console**, **Enable network COM+ access**, **Enable network DTC access**, **ASP.NET**, and **Internet Information service (IIS)**.

Exit the all the dialog boxes.

Configure DTC

Open command line prompt.

Run the following command line to start Component Services Console.

**%SystemRoot%\system32\com\comexp.msc**

Navigate to **Console Root->Component Services->Computers->My Computer**. Then, Right click on **My Computer** and click **Properties**.

Click **MSDTC**, and click **Security Configuratoin....**

Make sure the following option boxes checked.

**Network DTC Access**

**Allow Remote Clients**

**Allow Remote Administrations**

**Allow Inbound**

**Allow Outbound**

**Enable XA Transactions**

Install .NET Framework 2.0.

Install Windows SharePoint Services 2.0 with Service Pack 2



To download the Windows SharePoint Services 2.0 with Service Pack 2, visit <http://go.microsoft.com/fwlink/?LinkId=47398>.

On the **Download Windows SharePoint Services with Service Pack 2** page, under Download, click **Windows SharePoint Services with SP2**.

On the **Windows SharePoint Services with Service Pack 2** page, click **Download**.

Save the file to a location on your hard disk.

Locate the package you just downloaded and double-click **stsv2.exe**.

On the **End User License Agreement** screen, select **I accept the terms in the License Agreement**, and then click **Next**.

Under **Type of Installation**, ensure that **Server Farm** is selected, and then click **Next**.

On the **Summary** screen, click **Install**.

Close the **Configure Administrative Virtual Server** Web browser.

Run BizTalk Server 2006 setup program.

In Component Installation dialog box, select the following components to install:

**Business Activity Services**

**Enterprise Single Sign-On Administrators**

In Redistributable Prerequisites dialog box, select "Automatically install the redistributable prerequisites from a CAB file" option, and choose the pre-downloaded CAB file in the step 1.

## **To configure Trading Partner Management (TPM) server**

Configure Windows SharePoint Services.

Click **Start**, point to **Administrative Tools**, and then click **SharePoint Central Administration**.

Under **Application Pool**, ensure that **Create a new application pool** is selected.

In the **Application pool name** field, type **WSSAppPool**.

Under **Select a security account for this application pool**, ensure that **Configurable** is selected.

In the **User name** field, type **CONTOSO\wsssvc**.

In the **Password** field, type the domain account password, confirm your password, and then click **OK**.

Under **Security Configuration**, select **NTLM**.

The **Application Pool Changed** page loads and tells you that you must restart IIS.

Click **Start**, click **Run**, type **IISRESET**, and then click **OK**.

On the **Application Pool Changed** page, click **OK**. The **Configuration Database** page loads.

Under **Configuration Database**, in the **Database server** field, type the name of your server.

In the **SQL Server database name** field, type *WSSDB*.

Leave the rest of the default values, and then click **OK**. This creates the configuration database and you are redirected to the **Central Administration** Homepage.

Extend Default Web Site as a virtual server

On the **Central Administration Homepage**, under **Virtual Server Configuration**, select **Extend or upgrade virtual server**.

On the **Virtual Server List** page, under **Virtual Server List**, select **Default Web Site**.

On the **Extend Virtual Server** page, under **Provisioning Options**, select **Extend and create a content database**.

On the **Extend and Create Content Database** page, under **Create a new application pool**, in the **Application pool name** field, type *MyAppPool*.

Under **Select a security account for this application pool**, in the **User name** text box, type a domain account that will be used for this application pool.

In the **Password** text box, type the domain account password, and then confirm that password.

Under **Site Owner**, in the **E-mail text box**, type your e-mail address.

Under **Security Configuration**, select **NTLM**.

Leave the rest of the default values, and then click **OK**.

On the **Virtual Server Successfully Extended** page, select your new top-level Web site. This will open a new browser.

On the **Template Selection** page, select the **Team Site** template, and then click **OK**.

Close the Team Web Site you just created.

Close the SharePoint Central Administration tool.

Run Configuration.exe to start Microsoft BizTalk Server 2006 Configuration.

On the **Microsoft BizTalk Server 2006 Configuration page - Start** page,

Select **Custom configuration**.

Type E2E-SQL as **Database server name**.

Click **Configure**.

Click **Group** in the left pane, and in the right pane:

Select **Enable BizTalk Server Group on this computer** checkbox.

Select **Join an existing BizTalk Group** checkbox.

Click **SharePoint Adapter** in the left pane, and in the right pane:

Select **Enable Windows SharePoint Services Adapter on this computer** checkbox.

Type **CONTOSO\SharePoint Enabled Hosts** for the **Windows Group** of the **BizTalk SharePoint Adapter Enabled Hosts**.

Type your email in **Site e-mail** field.

Click **BAS** in the left pane, and in the right pane:

Select **Enable the Business Activity Services for this BizTalk Group** checkbox.

Click **BAS Security** in the left pane, and in the right pane:

Type the following windows groups and account for the **Windows groups/accounts**.

Role Name	Windows Group
BizTalk BAS Web Services Group	CONTOSO\BizTalk BAS Web Services Group
BizTalk BAS Users	CONTOSO\BizTalk BAS Users
BizTalk BAS Managers	CONTOSO\BizTalk BAS Managers
BizTalk BAS Administrators	CONTOSO\BizTalk BAS Administrators
BAS Enabled Hosts	CONTOSO\BizTalk BAS Enabled Hosts
BAS Site Owner	CONTOSO\basadmin

Type the following accounts for the Service Accounts

Name	Account
BAS Management Web Service Account	CONTOSO\basmgmtwebsvc
BAS Publishing Web Service Account	CONTOSO\baspblwebsvc
BAS Application Pool Account	CONTOSO\basapppool

Click **Apply Configuration** in the tool bar.

Click **Configure** to start configuration.

### To install legacy system.

The B2B solution assumes that the legacy system name is E2E-LEGACY.

Install SQL Server 2000 and SP4.

Select Mixed Mode (Windows Authentication and SQL Server Authentication)

Create btsb2bhub local user account. It must have the same name and passwords in the step "To create global domain security groups and user accounts that the B2B Solution will use".

## Installing the B2B Solution to Production Servers

The following steps describe how to install the B2B solution to the production servers prepared through "Preparing the Production Servers for the B2B Solution".

To extend the BAS

To configure the Windows SharePoint Service

To create TPMAccessWS web service

To set certificates

To configure the first BizTalk Server

To configure the HTTP Receive location to the first BizTalk Server

To configure the SMTP server on the first BizTalk Server

To configure the second BizTalk Server

To deploy the MSI to the first BizTalk Server

To deploy the MSI to the second BizTalk Server

To create the BizTalk Server Application from Administration Client

To import the MSI files for the B2B Solution from Administration Client

To import binding files and configure BizTalk Server applications from Administration Client

To create a SSO application, and set configuration values to the SSO store from Administration Client

To register BizTalk Servers on the BAS site

To create a profile using the BAS site

To create trading partners using the BAS site

To create agreements using the BAS site

To configure EDI Adapter

To configure the legacy system

## **Prerequisites**

Before deploying the B2B Solution to the production servers, you have to finish Developer Machine Setup for the Business-to-Business Solution.

## **To extend the BAS**

Logon to E2E-ADMIN using CONTOSO\btsadmin account.

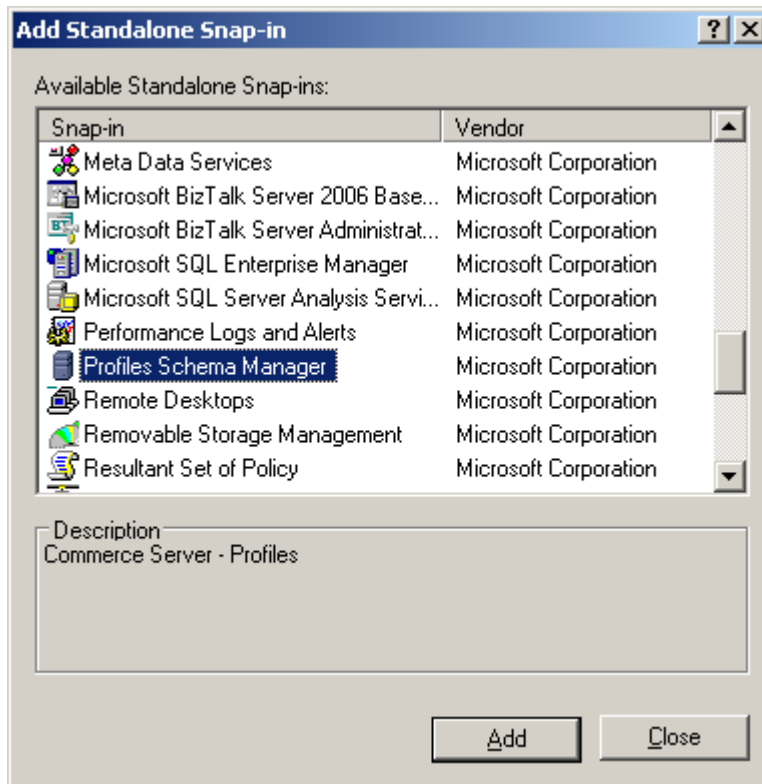
Copy the AdminClient folder you created in the How to Prepare the B2B Solution for Deployment to C:\.

Open Contact.SQL in the AdminClient folder using the SQL Query Analyzer. Run it against the BizTalkMgmtDB database.

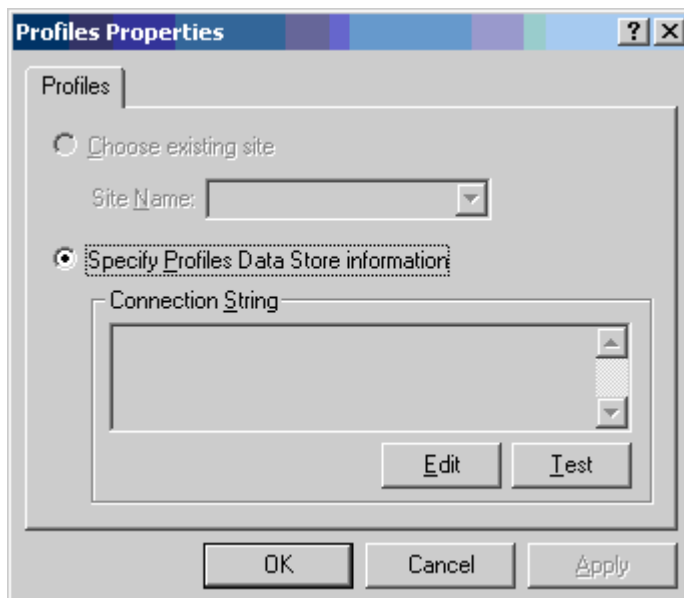
Logon to E2E-WEB using CONTOS\btsadmin account.

Copy the TPMServer folder you created in the to How to Prepare the B2B Solution for Deployment C:\.

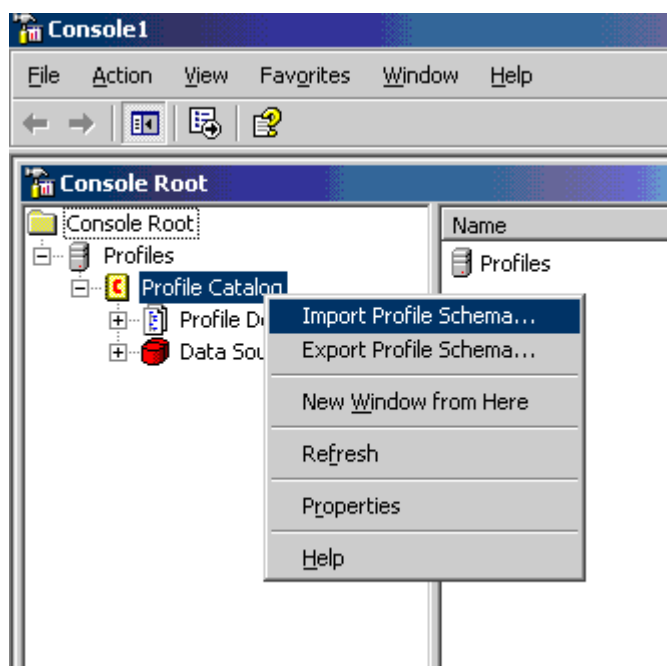
Run MMC.exe and add the **Profiles Schema Manager** snap-in.



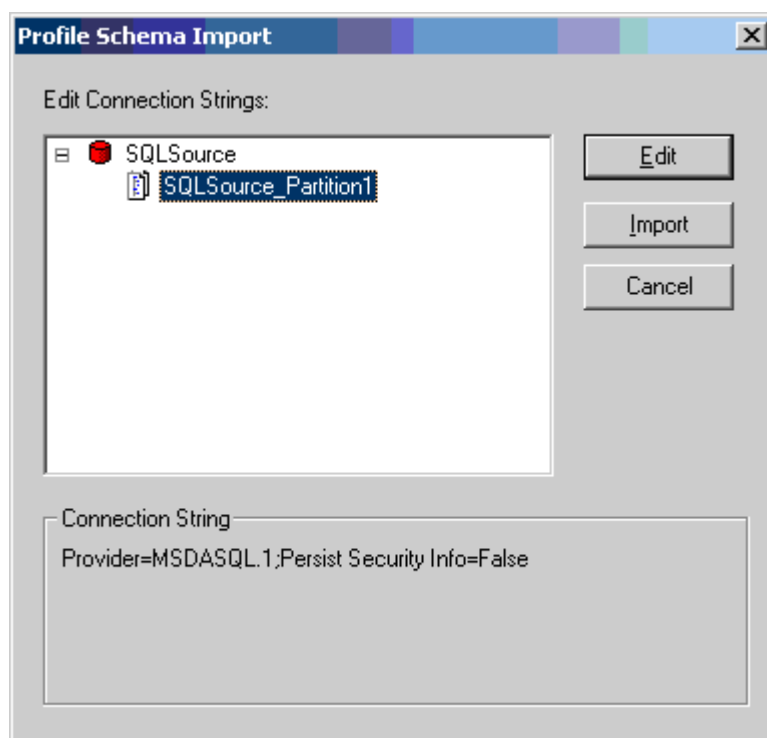
Right-click the **Profiles** node and select **Properties**. Click **Edit** and set your SQL connection properties to point to your TPM database. Click **OK** twice to exit the **Data Link Properties** and **Profile Properties** dialogs



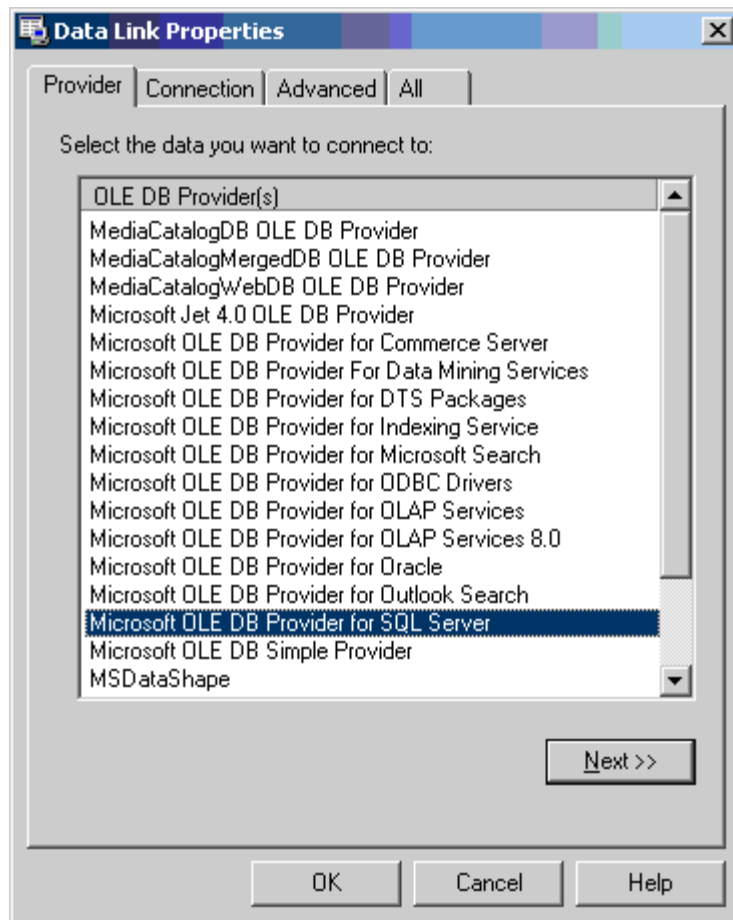
Import the extended profile schema from the file TPMUserSchemaNotEncrypted.xml in the TPMServer folder using the snap-in.



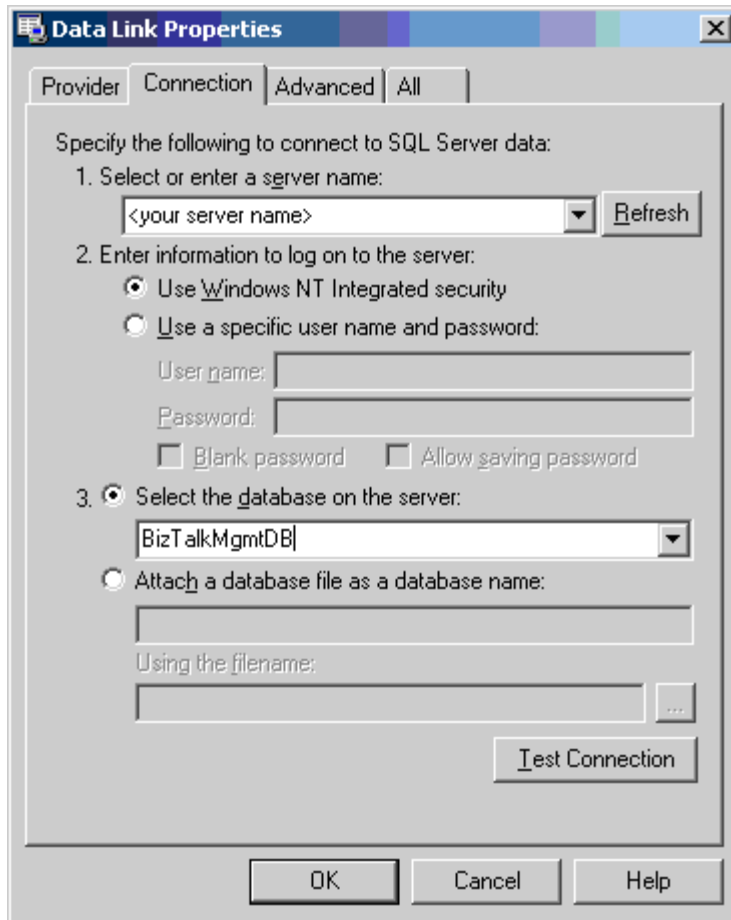
Import the schema by first selecting **SQLSource\_Partition1** and clicking **Edit**.



Make sure that the OLEDB Provider used is for SQL Server.







Click **Import** to import the extended profile schema. Check if the schema got extended using MMC.

Logon to E2E-ADMIN using CONTOSO\btsadmin account.

In Internet Explorer, in the **Address** box, type the URL of the Business Activity Services site, and then press ENTER. In this walkthrough the URL will be **http://E2E-WEB/sites/BASSite/**.

Type **CONTOSO\btsadmin** and the password in the authentication dialog box.

Upload the extended InfoPath form in the AdminClient folder:

- Select **Documents and Lists**

- Select the **Templates** folder.

- Click the **Upload Document** menu.

- Upload the file PartnerProfileAdmin.xsn.

- Repeat the procedure to upload the file PartnerProfileSelf.xsn.

Repeat the procedure to upload the file PartnerProfileStandard.xsn.

Backup the Partner.xsd and PartnerNoWsdL.xsd schemas in the %BTSInstallDir%\Business Activity Services\TPM\Schemas folder. Copy Schemas from TPM\Schemas in the B2B scenarios folder to the %BTSInstallDir%\Business Activity Services\TPM\Schemas folder.

### To configure the Windows SharePoint Service

Logon to E2E-WEB using CONTOSO\btsadmin account.

Remove Root from the list of SharePoint managed paths, as follows:

Open the SharePoint Central Administration site (Start > Programs > Administrative Tools > SharePoint Central Administration).

Select **Configure virtual server settings**.

Select Default Web Site.

Select Define Managed paths.

Under Included Paths, select Root and then click Remove selected paths.

Perform an IISReset.

Open **Command Prompt**. Click **Start**, click **Run** type **cmd**, and then click **OK**.

At the command prompt, change the current directory to %SystemDrive%\Inetpub\AdminScripts folder. Then, run the following command line to force IIS to use NTLM as authentication mechanism.

**cscript adsutil.vbs set W3SVC/1/ROOT/NTAuthenticationProviders "NTLM"**

### To create TPMAccessWS web service

Logon to E2E-WEB using CONTOSO\btsadmin account.

Copy the TPMAccessWS in the the TPMServer to the C:\Program Files\B2B.

Create the virtual directory TPMAccessWS.

Open IIS Manager (Start > Programs > Administrator Tools > Internet Information Services (IIS) Manager).

Expand Web Sites and right-click the **Default Web Site**.

Choose **New -> Virtual directory**

Click **Next** in the Virtual Directory Creation Wizard.

Type "TPMAccessWS" as **Alias** filed and click **Next**.

Select C:\Program Files\B2B\TPMAccessWS as **Path** field and click **Next**.

Check **Run scripts (such as ASP)**.

Click **Next** and **Finish** to exit.

Right-click TPMAccessWS and click **Properties**.

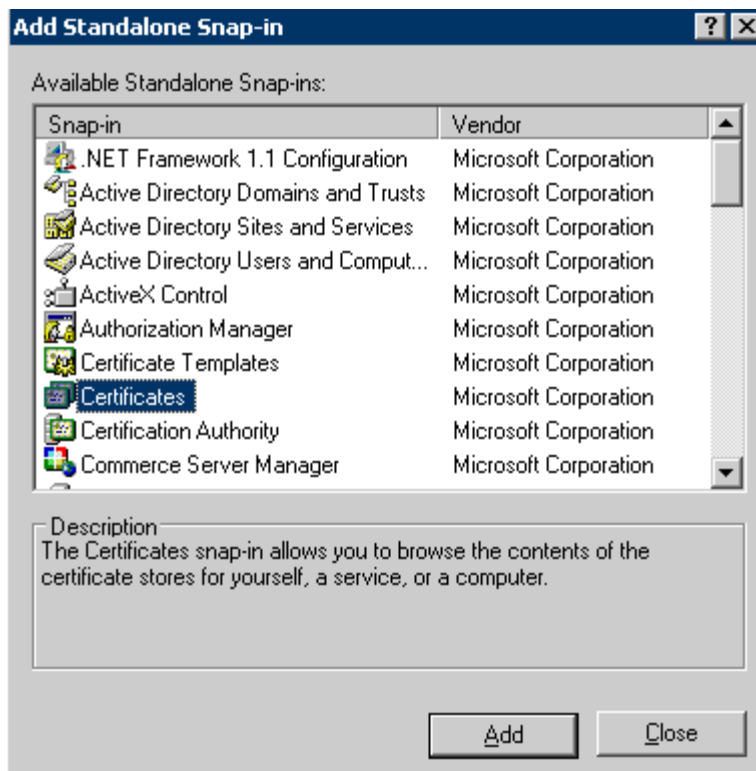
Select the **Application Pool** used by TPPubWS virtual directory.

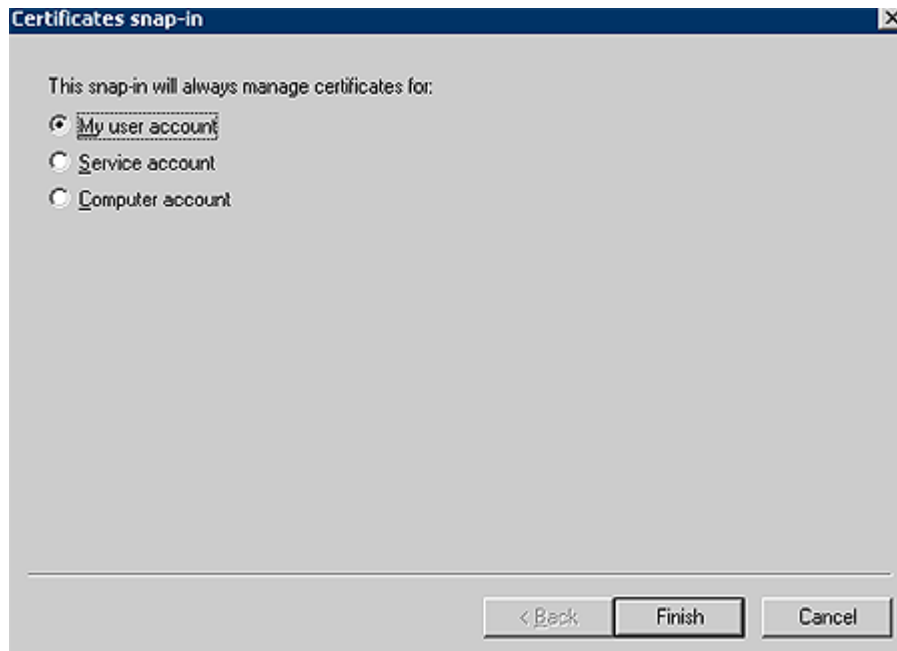
Make sure that users accessing the Web service are in the proper group ("BizTalk BAS Web Services Group").

### To set certificates

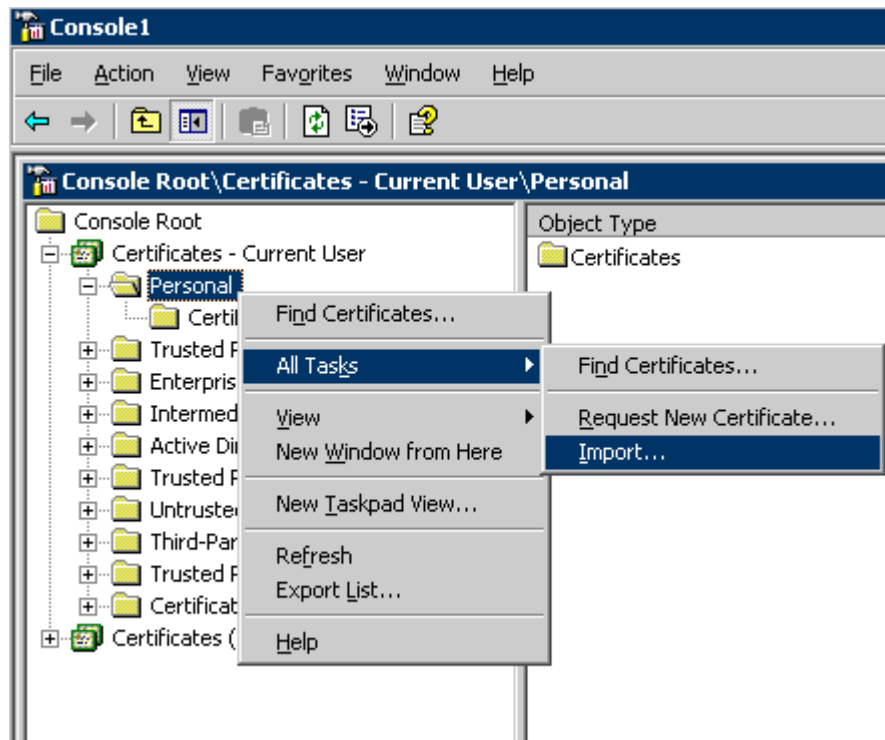
Logon to E2E-ADMIN using CONTOSO\btsadmin account.

Run MMC.exe and add the Certificates snap-in using **My user account**.

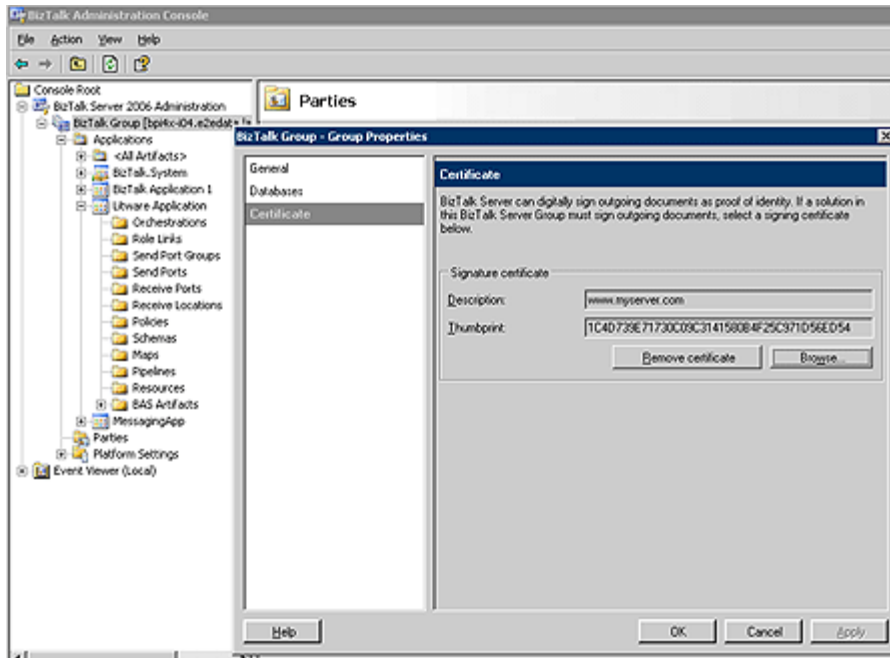




Import the myserver.pfx file in the AdminClient folder in Current user's "**Personal**" store, and accept the default values. Password is blank.

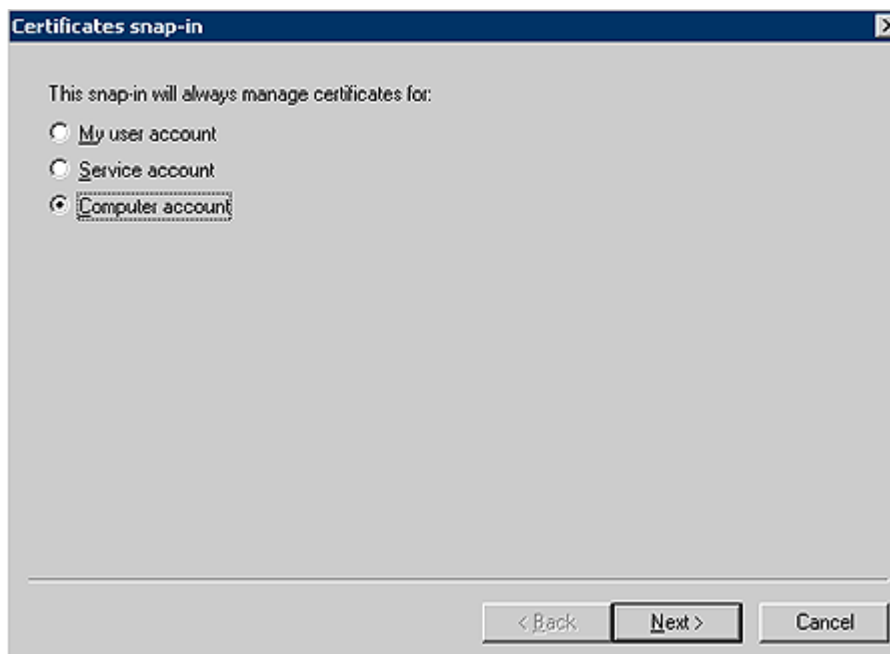


Open the BizTalk Administration console and right-click **BizTalk Group**. Choose **Properties** -> **Certificates** and click **Browse** to associate the imported certificate www.myserver.com with BizTalk Group.

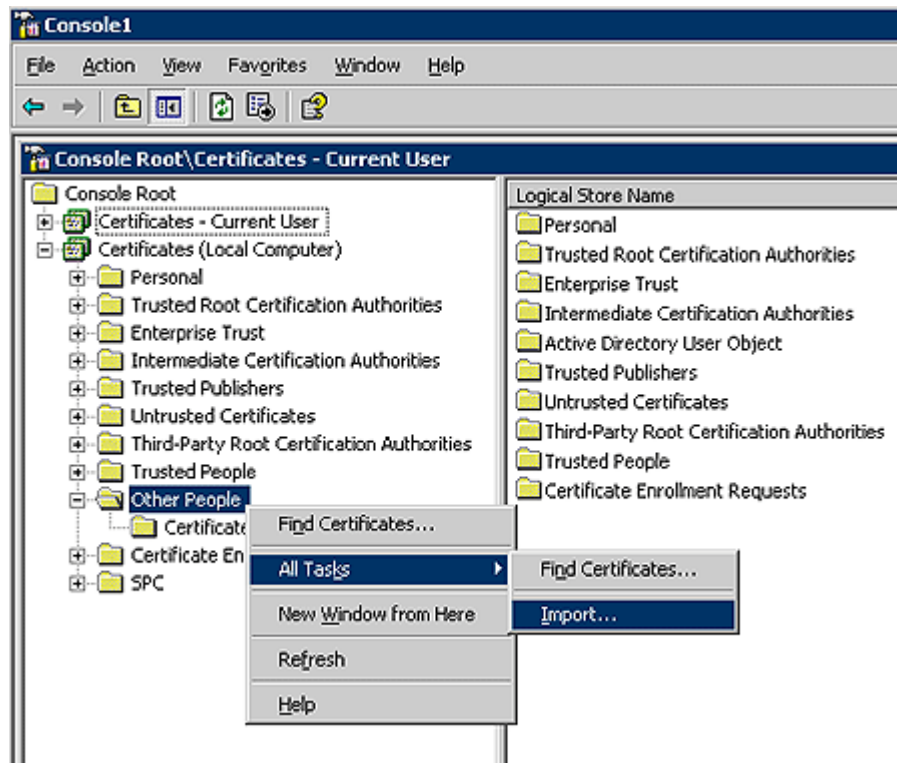


Logon to E2E-WEB using CONTOSO\btsadmin account

Run MMC.exe and add the Certificates snap-in using **Computer account**.



Import the myserver.pfx file in Local Computer "Other People" store. If you cannot find "**Other People**" store, use certmgr.exe which comes as part of the .NET Framework.



Logon to E2E-BTS1 using CONTOSO\btsadmin account

Copy the BizTalk Server folder you created to C:\.

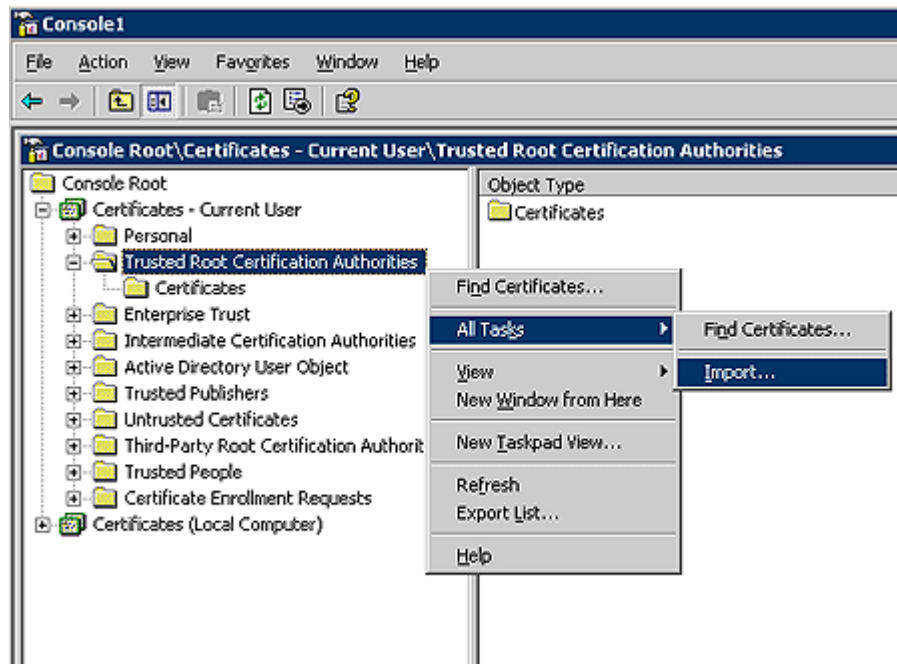
Run MMC.exe and add the Certificates snap-in using **Computer account**.

Import the myserver.pfx file in Local Computer "**Other People**" store.

Logoff from E2E-BTS1.

Logon to E2E-BTS1 again using CONTOSO\btsfileadapter.

Import the myserver.pfx file in the BizTalkServer folder in Current user's "**Personal**" and "**Trusted Root Certificate Authorities**" store, and accept the default values. Password is blank.



### To configure the first BizTalk Server

Logon to E2E-BTS1 again using CONTOSO\btsadmin.

Open **Command Prompt**. Click **Start**, click **Run** type **cmd**, and then click **OK**.

At the command prompt, navigate to the BizTalkServer folder and run Locations.cmd to create the location.

Make sure that read/write permissions are set on the access control lists (ACLs) for the receive location folders. The CONTOSO\btsfileadapter must have read/write access for the following receive location folders:

\BrokerNew

\BrokerUpdate

\EDIIn

\EDIOut

\GetOrder

\GetProfile

\UpdateUserBASRL

\Vendor1

Vendor2

Start the Registry Editor tool (Regedit.exe).

Expand to HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\ in the left panel to locate the EventLog folder icon at the preceding registry path.

Right-click the EventLog folder, and then click **Permissions**.

In the Permission for Eventlog dialog box, click **Add**.

Add CONTOSO\BizTalk CableOrder Host Users and CONTOSO\BizTalk OrderBroker Host Users groups.

Select **Allow** for **Full Control**.

Exit Regedit.exe.

Configure MSDTC for SQL Adapter to access to the remote SQL Server.

Make sure to enable Network DTC Access.

Check **no authentication required** check box on Security Configuration dialog box for MSDTC.

### To configure the HTTP Receive location to the first BizTalk Server

Logon to E2E-BTS1 again using CONTOSO\btsadmin.

Add CONTOSO\BizTalk SOAP Adapter Isolated Host Users to local IIS\_WPG group.

Run Internet Information Services (IIS) Manager.

In the IIS Manager, right-click **Web Service Extensions**, and then click **Add a new Web service extension**.

In the **New Web Service Extension** dialog box, do the following

Use this	To do this
<b>Extension Name</b>	Type the name of the new Web service extension.
<b>Required files</b>	Click <b>Add</b> , and then click <b>Browse</b> . Browse to the <drive>:\Program Files\Microsoft BizTalk Server 2006\HttpReceive directory, select <b>BTSHTTPReceive.dll</b> , click <b>Open</b> , and then click <b>OK</b> .
<b>Set extension status to</b>	Select this check box.



<b>Allowed</b>	
----------------	--

Click **OK**.

Right-click **Application Pools**, point to **New**, and then click **Application pool**.

In the **Add New Application Pool** dialog box, in the **Application pool ID** box, type a name for the application pool, and then click **OK**.

The new application pool appears in the list of **Application Pools**.

Right-click the new application pool, and then click **Properties**.

In the *<Application Pool name>* **Properties** dialog box, on the **Identity** tab, do the following.

Use this	To do this
<b>Configurable</b>	Select this property.
<b>User name</b>	Type CONTOSO\btshttpisolated
<b>Password</b>	Type the password for the user name.

Click **OK**.

Expand the **Web Sites** node, right-click the **Default Web Site** node, point to **New**, and then click **Virtual Directory**.

In the Virtual Directory Creation Wizard, on the **Welcome** page, click **Next**.

On the **Virtual Directory Alias** page, in the **Alias** box, type **HWSMessages** in **Alias** box, and then click **Next**.

On the **Web Site Content Directory** page, click **Browse**.

In the **Browse For Folder** dialog box, navigate to the *<drive>:\Program Files\Microsoft BizTalk Server 2006\HttpReceive* directory, click **OK**, and then click **Next**.

On the **Access Permissions** page, select **Read** and **Execute**, clear all other check boxes, and then click **Next**.

On the **Completion** page, click **Finish** to close and complete the wizard.

The new virtual directory appears under the list of Default Web Sites in Internet Information Services (IIS) Manager.

Right-click the virtual directory, and then click **Properties**.

In the <Virtual Directory> **Properties** dialog box, on the **Virtual Directory** tab, do the following.

Use this	To do this
<b>Execute Permissions</b>	Ensure that the property is set to <b>Scripts and Executables</b> .
<b>Application Pool</b>	Select the new application pool created earlier in this procedure.

Click **OK**.

On the **File** menu, click **Exit**.

### To configure the SMTP server on the first BizTalk Server

Logon to E2E-BTS1 again using CONTOSO\btsadmin.

Run Internet Information Services (IIS) Manager.

In the IIS Manager, right-click **Default SMTP Virtual Server**, and then click **Properties**.

In the **Default SMTP Virtual Server Properties** dialog box, on the **Access** tab, click **Relay**.

Check **all except the list below** in the **Relay Restriction** dialog box.

Exit all the dialog boxes.

### To configure the second BizTalk Server

Logon to E2E-BTS2 again using CONTOSO\btsadmin.

Follow the same steps as the previous instructions for the first BizTalk Server.

### To deploy the MSI to the first BizTalk Server

Logon to E2E-BTS1 again using CONTOSO\btsadmin.

Run BTSScn.B2B.Hub.msi in the BizTalkServer folder.

Type C:\Program Files\B2B\B2BHubApp\ for the folder name on the **Select Installation Folder** dialog box.

### To deploy the MSI to the second BizTalk Server

Logon to E2E-BTS2 again using CONTOSO\btsadmin.

Run BTSScn.B2B.Hub.msi in the BizTalkServer folder.

Type C:\Program Files\B2B\B2BHubApp\ for the folder name on the **Select Installation Folder** dialog box. To create the BizTalk Server Application from Administration Client

Logon to E2E-ADMIN using CONTOSO\btsadmin account.

Click **Start**, click **All Programs**, click **Microsoft BizTalk Server 2006**, and then click **BizTalk Server Administration**.

Click Connect to an existing group on the right pane.

### To import the MSI files for the B2B Solution from Administration Client

Open command line prompt, and then navigate to the C:\AdminClient folder

Run the following command line to import CableOrderApp.msi file as following

```
btstask ImportApp -Package:BTSScn.B2B.Hub.msi -  
ApplicatonName:BTSScn.B2B.Hub -Server:E2E-SQL -Database:BizTalkMgmtDb
```

### To import binding files, and configure BizTalk Server applications from Administration Client.

Open the BTSScn.B2B.Hub.BindingInfo.xml file using notepad.exe.

Edit the BTSScn.B2B.Hub.BindingInfo.xml file to change the host node for the Microsoft.Samples.BizTalk.Litware.Orchestrations.CheckOrderExists orchestration as following

Edit the BTSScn.B2B.Hub.BindingInfo.xml file to change the host node for the Microsoft.Samples.BizTalk.Litware.Orchestrations.ConfirmOrderWithBroker orchestration as following

Edit the BTSScn.B2B.Hub.BindingInfo.xml file to change the host node for the Microsoft.Samples.BizTalk.Litware.Orchestrations.ConfirmOrderWithVendors orchestration as following

Edit the BTSScn.B2B.Hub.BindingInfo.xml file to change the host node for the Microsoft.Samples.BizTalk.Litware.Orchestrations.EDIXmlToBasket orchestration as following

Edit the BTSScn.B2B.Hub.BindingInfo.xml file to change the host node for the Microsoft.Samples.BizTalk.Litware.Orchestrations.GetOrder orchestration as following

Edit the BTSScn.B2B.Hub.BindingInfo.xml file to change the host node for the Microsoft.Samples.BizTalk.Litware.Orchestrations.GetProfile orchestration as following

Edit the BTSScn.B2B.Hub.BindingInfo.xml file to change the host node for the Microsoft.Samples.BizTalk.Litware.Orchestrations.GetPurchaseOrder orchestration as following

Edit the BTSScn.B2B.Hub.BindingInfo.xml file to change the host node for the Microsoft.Samples.BizTalk.Litware.Orchestrations.NewOrder orchestration as following

Edit the BTSScn.B2B.Hub.BindingInfo.xml file to change the host node for the Microsoft.Samples.BizTalk.Litware.Orchestrations.UpdateOrder orchestration as following

Edit the BTSScn.B2B.Hub.BindingInfo.xml file to change the host node for the Microsoft.Samples.BizTalk.Litware.Orchestrations.UpdateUserProfileBAS orchestration as following

Edit the BTSScn.B2B.Hub.BindingInfo.xml file to change the host node for the Microsoft.Samples.BizTalk.Litware.Orchestrations.GetOrder orchestration as following

Edit the BTSScn.B2B.Hub.BindingInfo.xml file to change the host node for the Microsoft.Samples.BizTalk.Litware.Orchestrations.IsOrderExist orchestration as following

Edit the BTSScn.B2B.Hub.BindingInfo.xml file to change the host node for the Microsoft.Samples.BizTalk.Litware.Orchestrations.NewOrder orchestration as following

Edit the BTSScn.B2B.Hub.BindingInfo.xml file to change the host node for the Microsoft.Samples.BizTalk.Litware.Orchestrations.UpdateOrder orchestration as following

Edit the BTSScn.B2B.Hub.BindingInfo.xml file to change all the RecevieHandler nodes for the File Adapter as following

Edit the BTSScn.B2B.Hub.BindingInfo.xml file to change all the SendHandler nodes for the File Adapter as following

Edit the BTSScn.B2B.Hub.BindingInfo.xml file to change all the SendHandler nodes for the HTTP Adapter as following

Edit the BTSScn.B2B.Hub.BindingInfo.xml file to change all the SendHandler nodes for the SOAP Adapter as following

Edit the BTSScn.B2B.Hub.BindingInfo.xml file to change all the ReceiveHandler nodes for the HTTP Adapter as following

Edit the BTSScn.B2B.Hub.BindingInfo.xml file to change all the ReceiveHandler nodes for the EDI Adapter as following

In the **BizTalk Server Administration** Console, navigate to **Console Root->BizTalk Server 2006 Administration->BizTalk Group->Applications**. Right-click BTSScn.B2B.Hub, click **Import->Bindings**, select . BTSScn.B2B.Hub.BindingInfo in **File Name**, and then click **Open**.

In the **BizTalk Server Administration** Console, navigate to **Console Root->BizTalk Server 2006 Administration->BizTalk Group->Platform Settings->Adapters**.

Click **SMTP**, and then double-click the send handler for SMTP Adapter in the right pane. Change the **Host name** to SMTPAdapter you just created.

Click **Windows SharePoint Services**, click the send handler for SharePoint Adapter in the right pane. Change the **Host name** to SharePointAdapter.

Click the receive handler for SharePoint Adapter in the right pane, change the **Host name** to SharePointAdapter.

In the BizTalk Server Administration Console, navigate to **Console Root->BizTalk Server 2006 Administration->BizTalk Group->Applications**. Click BTSScn.B2B.Hub in the left pane, and then click **Send Ports**. Right-click TPMWrapperAccess\_SOAP in the right pane, click **Properties**, and then click **Configure** in the **Send Properties** dialog box.

Type **http://E2E-WEB/TPMAccessWS/TPMAccess.asmx** in the **Web service URL** box.

Exit all the dialog boxes.

### **To create a SSO application, and set configuration values to the SSO store from Administration Client.**

Open the LitwareSSOConfiguration.xml in the AdminClient folder with notepad.exe, and then modify it as following:

Open the CreateLitwareApplication.cmd in the AdminClient folder with notepad.exe, and then modify it as following:

Open **Command Prompt**. Click **Start**, click **Run** type **cmd**, and then click **OK**.

At the command prompt, type **set path=%path%;"%ProgramFiles%\Common Files\Enterprise Single Sign-On"**, and then press ENTER.

At the command prompt, change the current directory to C:\AdminClient, and then run the following command line to set the SSO server name.

**ssomanage -serverall E2E-BTS1**

At the command prompt, run **CreateSouthridgeVideoApplication.cmd**.

### **To register BizTalk Servers on the BAS site**

Logon to E2E-ADMIN using CONTOSO\btsadmin account.

In Internet Explorer, in the **Address** box, type the URL of the Business Activity Services site, and then press ENTER. In this walkthrough the URL will be **http://E2E-WEB/sites/BASSite/**.

Type **CONTOSO\btsadmin** and the password in the authentication dialog box.

In your browser, navigate to **Trading Partner Management Tools -> BizTalk Servers -> Register BizTalk Server**.

Enter the registration information and Refresh Host Lists.

Enter the registration name you want.

Type **E2E-SQL** as BizTalk Management Database Server Name

Enter BizTalk Management Database. The default name of BizTalk Management Database is **BizTalkMgmtDb**.

Refresh Host Lists. It will retrieve the list of BizTalk Hosts from servers. The host configured as Windows SharePoint Services Adapter Receive Handler will be shown for **Outbox Receive Location Host**. The host configured as SOAP Adapter Send Handler will be shown for **Parameter Services Host**.

From the drop-down list, select a host to use for **Outbox Receive Location Host** and **Parameter Services Host**.

Save and close.

### To create a profile using the BAS site

Ligon to E2E-ADMIN using CONTOSO\btsadmin account.

Open the MyProfile.xml in the AdminClient folder, and then change the following elements to point to the correct Web Service server.

```
<?mso-infoPathSolution href="http://E2E-  
WEB/sites/BASSite/Templates/PartnerProfileSelf.xsn" ...?>  
  
<?ms-tpm url=http://E2E-WEB/TPPubWS/TPPubWS.asmx?>  
  
<?ms-tpmgmtws url=http://E2E-WEB/TPMgmtWS/TPMgmtWS.asmx?>
```

Using InfoPath, open MyProfile, and then change the email address someone@microsoft.com in the Contacts section to the correct email address to get notifications. You can do this by selecting **Contacts** in the InfoPath toolbar and then modifying the **E-mail address: value**.

**Microsoft Office InfoPath 2003**

File Edit View Insert Format Tools Table Help

General Information Addresses **Contacts** Print View Submit Profile Help

**My Profile** Microsoft BizTalk Server

**Contacts**  
Use the Contacts tab to enter contact information that you want to share with a partner. You can enter multiple sets of contact information in each profile. For example, you could add both a shipping contact and a billing contact.

Contact name: MyContact

Contact description: MyContact

User Name: MyContact

Password: [password box]

Contact type: MyContactType

Telephone number: 999-999-9999 Telephone description: 999-999-9999

☒ Insert telephone

Fax number: 999-999-9999 Fax description: 999-999-9999

☒ Insert Fax

E-mail address: someone@microsoft.com E-mail description: someone@microsoft.com

☒ Insert e-mail

☒ Insert contact

Submit the profile.

### To create trading partners using the BAS site

Logon to E2E-ADMIN using CONTOSO\btsadmin account.

Open the TestPartnerExtended.xml in the AdminClient folder using notepad.exe, and then change the following elements to point to the correct Web Service server.

```
<?mso-infoPathSolution href="http://E2E-WEB/sites/BASSite/Templates/PartnerProfileAdmin.xsn" ...?>
```

```
<?ms-tpm url=http://E2E-WEB/TPPubWS/TPPubWS.asmx?>
```

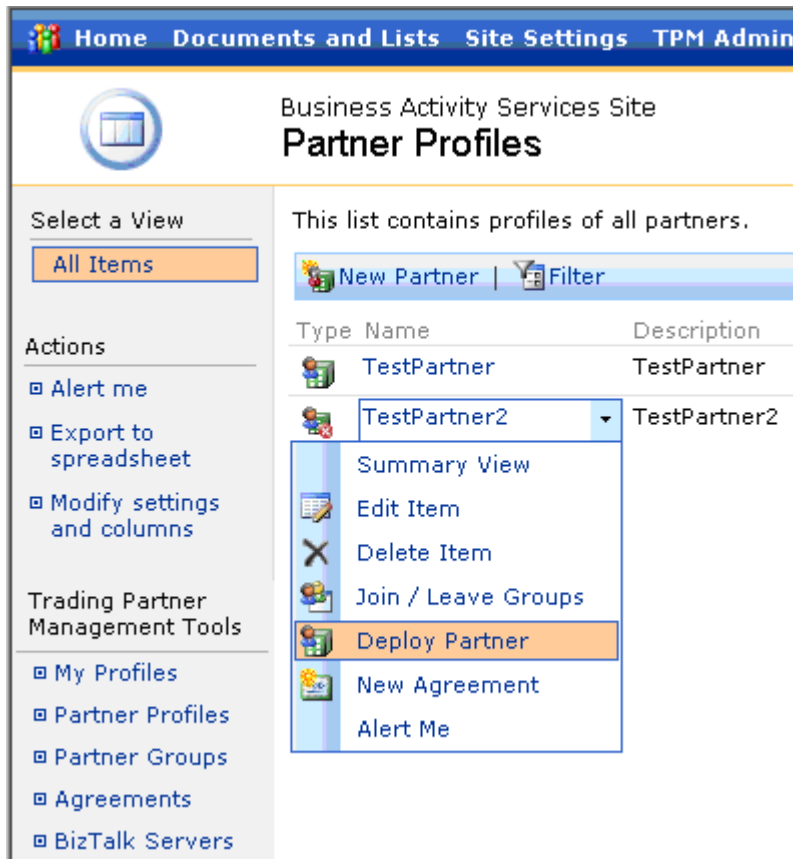
```
<?ms-tpmgmtws url=http://E2E-WEB/TPMgmtWS/TPMgmtWS.asmx?>
```

Using InfoPath, open the TestPartnerExtended.xml. Change the email address someone@microsoft.com in the Contacts section to the correct email address to get notifications. You can do this by selecting **Contacts** in the InfoPath toolbar and then modifying the **E-mail address: value**.

Submit the profile by selecting the **Submit Profile** in the InfoPath toolbar.

Open the BAS site (Start > Programs > Microsoft BizTalk Server 2006 > Business Activity Services Site), and click **the Partner Profiles** link under **Trading Partner Management Tools**. Expand the TestPartner drop down control, choose **Deploy Partner**. Then, select

registration name you typed in connecting the BAS Site to BizTalk Server, and choose **Deploy**.



Open the TestPartner2Extended.xml in the AdminClient folder using notepad.exe, and then change the following elements to point to the correct Web Service server.

```
<?mso-infoPathSolution href="http://E2E-
WEB/sites/BASSite/Templates/PartnerProfileSelf.xsn" ...?>
```

```
<?ms-tpm url=http://E2E-WEB/TPPubWS/TPPubWS.asmx?>
```

```
<?ms-tpmgmtws url=http://E2E-WEB/TPMgmtWS/TPMgmtWS.asmx?>
```

Using InfoPath, open the TestPartner2Extended.xml. Change the email address someone@microsoft.com in the Contacts section to the correct email address to get notifications. You can do this by selecting **Contacts** in the InfoPath toolbar and then modifying the **E-mail address: value**.

Submit and deploy the partner profile as same as TestPartnerExtended.xml.

### To create agreements using the BAS site

Logon to E2E-ADMIN using CONTOSO\btsadmin account.



Open the **Agreement.xml** in the **AdminClient** folder using **notepad.exe**, and then change the following elements to point to the correct Web Service server.

```
<?mso-infoPathSolution  
  href="http://localhost/sites/BASSite/Templates/Agreement.xsn"...?>
```

```
<?ms-tpm url=http://E2E-WEB/TPPubWS/TPPubWS.asmx?>
```

```
<?ms-tpmgmtws url=http://E2E-WEB/TPMgmtWS/TPMgmtWS.asmx?>
```

Create a new agreement between **TestPartner** and **MyProfile** by using **InfoPath** to open the **Agreement.xml**. Once the agreement is opened in **InfoPath**, click **Summit Agreement** button in the tool box.

The screenshot shows the Microsoft Office InfoPath 2003 application window titled "Agreement.xml - Microsoft Office InfoPath 2003". The menu bar includes File, Edit, View, Insert, Format, Tools, Table, and Help. The toolbar contains various icons for file operations, editing, and navigation. Below the toolbar, there are tabs for "General Information", "Addendums", "Legal Terms", "Print View", "Submit Agreement", and "Help". The "Submit Agreement" tab is currently selected. The main content area displays the "Agreement" form, which is part of "Business Activity Services". The form includes a "General Information" section with the following fields: "Agreement name" (containing "TestPartner"), "Description" (containing "Agreement between MyProfile and TestPartner"), "Agreement date" (containing "3/28/2005"), and "Select self profile". There is also an "Insert duration" button. The "Submit Agreement" button is located in the top right corner of the form area.

Activate the agreement as the following figure. When you activate the agreement, the agreement will be opened in **InfoPath**. Assign all seven Operation ports to **VendorA\_FILE**. When finished, select **Submit Mapping**.

The screenshot shows the 'Business Activity Services Site Agreements' page. The top navigation bar includes 'Home', 'Documents and Lists', 'Site Settings', and 'TPM Admin'. The page title is 'Business Activity Services Site Agreements'. On the left, there is a sidebar with 'Select a View' (All Items), 'Actions' (Alert me, Export to spreadsheet, Modify settings and columns), and 'Trading Partner Management Tools' (My Profiles, Partner Profiles, Partner Groups, Agreements, BizTalk Servers). The main content area shows a table of agreements with columns 'Type Name' and 'Description'. A context menu is open over the 'TestPartner' agreement, showing options: 'Summary View', 'Edit Item', 'Delete Item', 'Activate Agreement' (highlighted), and 'Alert Me'. A 'Select Port -- Web Page Dialog' box is overlaid on the bottom, showing a list of ports: 'VendorA FILE' and 'EDISP'. The dialog has 'OK' and 'Cancel' buttons.

Home Documents and Lists Site Settings TPM Admin

Business Activity Services Site  
**Agreements**

Select a View  
All Items

Actions  
☐ Alert me  
☐ Export to spreadsheet  
☐ Modify settings and columns

Trading Partner Management Tools  
☐ My Profiles  
☐ Partner Profiles  
☐ Partner Groups  
☐ Agreements  
☐ BizTalk Servers

This list contains all the agreements.

New Agreement | Filter

Type Name	Description
TestPartner	Agreement
Summary View	Agreement
Edit Item	
Delete Item	
Activate Agreement	
Alert Me	

Select Port -- Web Page Dialog

Select Port:

VendorA FILE  
EDISP

OK Cancel

Open the **Agreement2.xml** in the **AdminClient** folder using **notepad.exe**, and then change the following elements to point to the correct Web Service server.

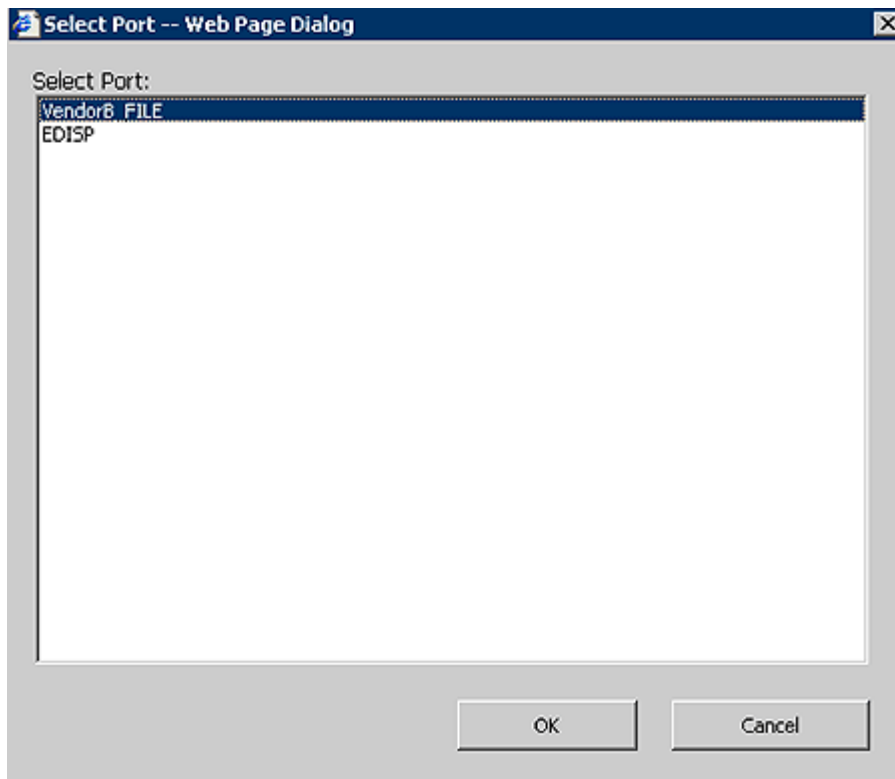
```
<?mso-infoPathSolution  
  href="http://localhost/sites/BASSite/Templates/Agreement.xsn"...?>
```

```
<?ms-tpm url=http://E2E-WEB/TPPubWS/TPPubWS.asmx?>
```

```
<?ms-tpmgmtws url=http://E2E-WEB/TPMgmtWS/TPMgmtWS.asmx?>
```

Create a new agreement between **TestPartner2** and **MyProfile** by using **InfoPath** to open **Agreement2.xml**. Once the agreement is opened in **InfoPath**, click **Summit Agreement** button in the tool box.

Activate the agreement. When you activate the agreement, the agreement will be opened in **InfoPath**. Assign all seven Operation ports to **VendorB\_FILE**. When finished, select **Submit Mapping**.



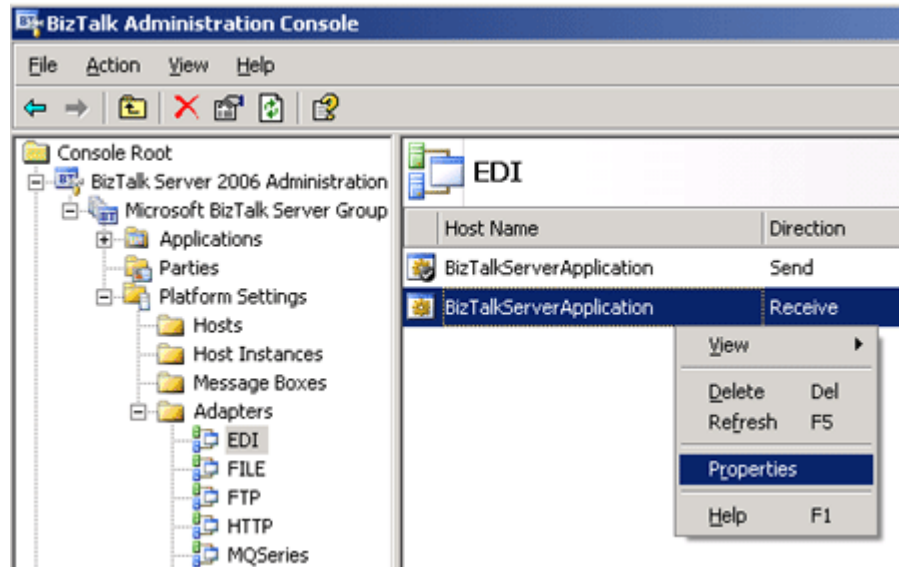
### To configure EDI Adapter

Logon to **E2E-ADMIN** again using **CONTOSO\btsadmin**.

Click **Start**, click **All Programs**, click **Microsoft BizTalk Server 2006**, and then click **BizTalk Server Administration**.

In the BizTalk Server Administration console, navigate to **BizTalk Group > Platform Settings > Adapters > EDI**.

Open the **Receive Handler Properties** window.



Click **Properties**.

Navigate to **FileSystem > Account**.

Under the **Folder** field, enter:

**C:\Litware\EDI\In**

Logon to E2E-BTS2 again using CONTOSO\btsadmin.

Grant CONTOSO\ btsedisubsystemsvc the Full Control permission to the folders under BPM folders.

Logon to E2E-BTS2 again using CONTOSO\btsadmin.

Grant CONTOSO\ btsedisubsystemsvc the Full Control permission to the folders under BPM folders.

Open **Command Prompt**. Click **Start**, click **Run** type **cmd**, and then click **OK**.

At the command prompt, navigate to the %ProgramFiles%\Microsoft BizTalk Server 2006\EDI\Adapter\EDI Schemas\x12\4010 folder.

Run the following command line to deploy the EDI Schema which the B2B Solution uses to the EDI database.

**xsd2edi 850Schema.xsd.**

Run the following command line to compile EDI Engine input file

**COMPEIF.**

### To configure the legacy system

Logon to E2E-LEGACY using local administrator user account.

Create OrderSystemDB database for logging.

Click **Start**, click **All Programs**, click **Microsoft SQL Server**, and then click **Query Analyzer**.

Type localhost in **SQL Server** box, and then click **OK**.

Open the Order.SQL in the LegacyServer folder, and then execute the script.

Exit SQL Query Analyzer.

Grant CONTOSO\btsb2bhub the permission to logon and execute the stored procedures of OrderSystemDB database.

Click **Start**, click **All Programs**, click **Microsoft SQL Server**, and then click **Enterprise Manager**.

In **Enterprise Manager**, in the left pane, expand **Microsoft SQL Servers**, expand **SQL Server Group**, expand **(LOCAL) Windows NT**, expand **Security**, right-click **Logins**, and then click **New Login**.

In the **General** tab, type E2E-LEGACY\btsb2bhub in the **Name** box

In the **Database Access** tab, select **Permit** for OrderSystemDB.

Click **OK** to exit the dialog box.

In the **Enterprise Manager**, in the left pane, expand **Databases**, expand OrderSystemDB, expand **Users**.

Right-click btsb2bhub in the right pane, click **Properties**, and then click **Permission** in the dialog box.

Check the **EXE** permissions for GetOrder, GetOrderStamp, GetVendorID, IsOrderExist, PlaceOrder, and Update Order stored procedures.

Exit all the dialog boxes.

Configure MSDTC for the B2BHubOrchestration to access the OrderSystemDB database.

Make sure to **enable Network DTC Access**.

Click **Start**, click **All Programs**, click **Administrative Tools**, and then click **Component Services**.

In the **Component Services** console, expand **Console Root**, expand **Component Services**, expand **Computers**, right-click **My Computer**, and then click **Properties**.

In the **MSDTC** tab in the dialog box, click **Security Configuration**.

Select the following options in the Security Configuration dialog box.

**Network DTC Access**

**Allow Remote Clients**

**Allow Remote Administration**

**Allow Inbound, Allow Outbound**

**No Authentication Required**

**Enable A Transactions**

Logon to E2E-SQL using CONTOSO\btsadmin.

Grant CONTOSO\btsb2bhub the permission to execute admsvr\_GetPartyByAliasNameValue stored procedure of the BizTalkMgmtDb.

Click **Start**, click **All Programs**, click **Microsoft SQL Server**, and then click **Enterprise Manager**.

In **Enterprise Manager**, in the left pane, expand **Microsoft SQL Servers**, expand **SQL Server Group**, expand **(LOCAL) Windows NT**, expand **Security**, right-click **Logins**, and then click **New Login**.

In the **General** tab, type CONTOSO\btsb2bhub in the **Name** box

In the **Database Access** tab, select **Permit** for the **BizTalkMgmtDb**.

Click **OK** to exit the dialog box.

In the **Enterprise Manager**, in the left pane, expand **Databases**, expand **BizTalkMgmtDb**, expand **Stored Procedures**.

Right-click admsvr\_GetPartyByAliasNameValue stored procedure in the right pane, click **Properties**, and then click **Permission** in the dialog box.

Check the **EXE** permissions for CONTOSO\btsb2bhub user account.

Exit all the dialog boxes.

Logon to E2E-BTS1 using CONTOSO\btsadmin.

Configure MSDTC for the B2BHubOrchestration to access the OrderSystemDB database.

Click **Start**, click **All Programs**, click **Administrative Tools**, and then click **Component Services**.

In the **Component Services** console, expand **Console Root**, expand **Component Services**, expand **Computers**, right-click **My Computer**, and then click **Properties**.

In the **MSDTC** tab in the dialog box, click **Security Configuration**.

Select the following options in the Security Configuration dialog box.

**Network DTC Access**

**Allow Remote Clients**

**Allow Remote Administration**

**Allow Inbound, Allow Outbound**

**No Authentication Required**

**Enable A Transactions**

Logon to E2E-BTS2 using CONTOSO\btsadmin.

Configure MSDTC for the B2BHubOrchestration to access the OrderSystemDB database as same as the E2E-BTS1.

## Running the B2B Solution on Production Servers

The following steps describe how to run and validate the Business-to-Business solution in the production environment.

To start BTSScn.B2B.Hub application

To start BizTalk Base EDI Service

To run and validate the B2B solution

### Prerequisites

Before running the B2B Solution with the production servers, you have to finish Installing the B2B Solution to Production Servers

### To start BTSScn.B2B.Hub application

Logon to E2E-ADMIN using CONTOSO\btsadmin account.

Click **Start**, click **All Programs**, click **Microsoft BizTalk Server 2006**, and then click **BizTalk Server Administration**.

In the **BizTalk Server Administration** console, expand **Console Root**, expand **BizTalk Group**, expand **Applications**, right-click BTSScn.B2B.Hub, click **Start**, and then click **Start**.

### To start BizTalk Base EDI Service

Logon to E2E-BTS1 using CONTOSO\btsadmin account.

Click **Start**, click **All Programs**, click **Administrative Tools**, and then click **Services**.

Make sure the **Status** of the **BizTalk Base EDI Service** is **Started**.

Logon to E2E-BTS2 using CONTOSO\btsadmin account.

Click **Start**, click **All Programs**, click **Administrative Tools**, and then click **Services**.

Make sure the **Status** of the **BizTalk Base EDI Service** is **Started**.

### To run and validate the B2B solution

Logon to E2E-BTS1 using CONTOSO\btsadmin account.

Creating new orders:

Drop Basket.xml in the BizTalkServer\SampleMessages folder into the C:\Litware\BrokerNew folder.

Check the Vendor1 folder for validation.

Check email notification (sent to the email address entered when creating a trading partner using the BAS site) for validation.

Update existing orders:

Create a new message named BasketUpdate.xml by copying the Basket.xml.

Modified the timestamp in the LastModified attribute of Basket element in the BasketUpdate.xml. You must not modify SoldTold and OrderGroupID attributes which identify the existing order you created in the previous step.

Create a signed message out of the BasketUpdate.xml. This walkthrough use BasketUpdateSigned.xml for the name of the signed message. For more information on how to sign messages, see Certificates and Signing Messages.

Drop the BasketUpdateSigned.xml into the BrokerUpdate folder.

Check the Vendor1 folder for validation.



Check email notification for validation.

Retrieve orders:

Create a signed message out of the OrderInquiry.xml in the BizTalkServer\SampleMessages folder. For this walkthrough, use GetOrderSigned.xml for the name of the signed message.

Drop GetOrderSigned.xml into the GetOrder folder.

Check the Vendor1 folder for validation.

Retrieve partner profiles:

Create a signed message out of the Partner.xml in the BizTalkServer\SampleMessages folder. For this walkthrough, use GetProfileSigned.xml for the name of the signed message.

Drop GetProfileSigned.xml into the GetProfile folder

Check the Vendor1 folder for validation.

Update partner profiles (file generated from GetProfileSigned.xml):

Use the result of RetrieveProfile and drop it into the UpdateUserBASRL folder.

Check the Vendor1 folder for validation.

Verify EDI:

Move the ContosoPickupInstance.edi message in the BizTalkServer\SampleMessages folder to the EDIIn folder.

Check the Vendor1 and Vendor2 folders for validation.

## How BizTalk Server Processes Large Messages

### What is a large message?

Unfortunately, the answer to this question isn't tied directly to a specific message size, but rather, depends on specific bottlenecks in your Microsoft BizTalk Server 2006 system. The problems associated with large messages can be divided into the following categories:

**Out of memory errors** Certain types of message processing - such as mapping, validation, and property promotion - load the entire message into memory. If the size of the message in memory exceeds available resources, then an out of memory error occurs. The size threshold for messages that fall into this category is much lower than the size threshold for messages that are not loaded into memory. For example, a 10 MB flat file that is parsed into XML and then mapped may grow by a factor of 10 or more to consume over 100 MB of memory,

whereas a 100 MB XML document that is not parsed or mapped may actually only consume 1 MB of memory as it is streamed to the MessageBox database.

**Performance problems for messages that are not loaded into memory** Messages that are not required to be loaded into memory are streamed to the MessageBox database using the .NET XmlReader interface. While they are not subject to the size limitations on messages that must be loaded into memory, there are some important factors that impact how BizTalk Server processes messages that are streamed to the MessageBox database.

### **Factors that affect the processing of large messages**

The original message size, message format, and type of message processing all affect how BizTalk Server processes large messages.

**Original message size** The size of the message received by BizTalk Server is the most visible indication of how large the message will be when it is processed by BizTalk Server. The original size of a message has a much greater impact on performance if the entire message is loaded into memory than if the message is streamed to the MessageBox database.

**Message format** Messages are received into BizTalk Server in one of two primary formats: XML files or flat files.

**XML files** In order for BizTalk Server to perform any processing on a message other than pass through routing, the message must be in the XML file format. If the files to be processed are received in XML format then they will retain their original size for the most part.

**Flat files** Flat files must be parsed into an XML format before BizTalk Server can perform any processing other than pass through routing. Parsing a flat file into an XML file can greatly increase the size of the file. Flat files do not contain XML tags with descriptive information about their data. XML files, on the other hand, wrap all of their data in descriptive XML tags. In some scenarios parsing can increase the size of a flat file by a factor of 10 or more, depending on how much descriptive data is contained in the XML tags for the file.

**Flat file documents wrapped in a single CDATA section node in an XML document** This type of document is a combination of both XML and flat file and can be problematic because BizTalk Server must load the entire wrapped flat file document into memory before processing it.

**Type of message processing** In BizTalk Server, there are two types of message processing: Routing only and Mapping. The performance variables tied to the type of message processing that is performed are message size and whether the message is loaded into memory.

**Routing only** If BizTalk Server is only used only for routing messages based upon promoted message properties, then the message is streamed into the MessageBox database using the .NET XmlReader interface, and message parts are not individually loaded into memory. In this scenario, out of memory errors are not an issue and the primary consideration is the amount of time that is required to write very large messages (over 100 MB) into the MessageBox database. The BizTalk Server development team has successfully tested the processing of messages up to 1 GB in size when performing routing only.

The primary factor that determines performance in this scenario is the **Large message fragment size** used to fragment the data into the database. The **Large message fragment size** is a configurable option on the **BizTalk Group Properties** configuration page and has a default value of 102400 bytes (100 KB). Increasing this value reduces the number of round trips required to stream a message into the MessageBox database.

**Mapping** Transforming a document with a map is a memory-intensive operation. When a document is transformed by a map, BizTalk Server passes the message stream to the .Net XslTransform class, which then loads the document into a .NET XPathDocument object for processing. Loading the document into the .NET XPathDocument can potentially expand the original file size in memory by a factor of 10 or more. This expansion may be more pronounced when mapping flat files because flat files must be parsed into XML before they can be transformed.

BizTalk Server 2006 significantly improves memory management for large documents by implementing a configurable message size threshold for loading documents into memory during transforms. Any message with a size below this threshold is handled in memory; any message with a size above this threshold is buffered to the file system to reduce memory requirements. The default message size threshold is 1 MB.

## Guidelines for processing large messages

Follow these guidelines to improve performance when processing large messages in BizTalk Server:

Adjust the message size threshold above which documents are buffered to the file system during mapping. To modify the size threshold, create a DWORD value named **TransformThreshold** at the following location in the BizTalk Server registry:

After you have created this value, enter a decimal value with the number of bytes to set the new threshold to. For example, enter a decimal value of 2097152 to increase the message size threshold to 2 MB (from the default of 1 MB). Increase this value on systems with a large amount of available memory to improve throughput. Buffering documents to disk conserves memory at a slight cost to overall throughput.

Minimize the use of maps in orchestrations:

If you are using a map to extract or set properties used with business logic in an orchestration, use distinguished fields or promoted properties instead. When extracting or setting values with a map the document is loaded into memory. When using distinguished fields or promoted properties, the Orchestration engine accesses the message context and does not load the document into memory.

If you are using a map to aggregate several fields into one field, use distinguished fields or promoted properties with an orchestration variable to accumulate the result set.

Adjust the **Large message fragment size** property exposed on the **BizTalk Group Properties** configuration page:

Increasing this value reduces the number of round trips required to stream a message into the MessageBox database and increases performance; however, this also increases the amount of available memory that is used.

When BizTalk Server writes a message to the MessageBox it does so under the context of a Microsoft Distributed Transaction Coordinator (MSDTC) transaction. If it takes longer than the default MSDTC transaction timeout of 60 minutes to write a message to the MessageBox, then the transaction times out, an error results, and the attempt to write the message fails and is rolled back. The **Large message fragment size** value should be increased enough to avoid this problem when processing very large messages. Depending on available memory, this value should be increased to a value between 1 MB and 10 MB (1048576 to 10485760 bytes).

Each message fragment in a message creates one or more SQL Server database locks against the MessageBox database. When the number of locks exceeds several hundred thousand, it is possible that SQL Server will start generating "out of lock" errors. If this problem occurs, increase the **Large message fragment size** to reduce the number of SQL Server database locks made against the MessageBox database.

Consider housing your MessageBox database on a 64-bit version of SQL Server if you are experiencing "out of lock" errors. The number of available locks is significantly higher on the 64-bit version of SQL Server.

## Development

This section contains information for developers who are tasked with creating BizTalk applications. Once these applications are created, they are usually managed by an administrator, using information found in the Operations section.

The topic **Developing BizTalk Server Applications** discusses the activities associated with developing a BizTalk Server application, which consists of such things as schema definitions, message mappings, receive and send pipelines, message subscriptions, and orchestrations.

The topic **Creating Custom Components** discusses the development of components which are used to modify or extend certain elements of BizTalk Server infrastructure, such as send or receive adapter components, send or receive pipeline components, and functoid components. These elements of infrastructure are used by BizTalk Server applications, but are not themselves parts of a BizTalk Server application.

### In This Section

- Developing BizTalk Server Applications
- Creating Custom Components
- Utilities in the SDK

## Developing BizTalk Server Applications

This section contains information for developers who are tasked with creating BizTalk projects. Projects are created using the BizTalk project System Design Environment, which allows you to design, organize, and build the various elements of BizTalk applications. The following sections describe this process in detail.

**In This Section**

- Developer Tools
- Using the BizTalk Messaging Engine
- Creating Schemas Using BizTalk Editor
- Managing Business Processes Using BizTalk Explorer
- Creating Maps Using BizTalk Mapper
- Creating Pipelines Using Pipeline Designer
- Creating Orchestrations Using Orchestration Designer
- Creating Business Rules Using Business Rules Editor
- Using Web Services
- International Considerations for Designing BizTalk Applications
- Deploying BizTalk Assemblies from Visual Studio into a BizTalk Application
- Creating a Single Sign-On Application
- Viewing Assemblies with the BizTalk Assembly Viewer

**Developer Tools**

The BizTalk project system is hosted in Microsoft® Visual Studio® 2005 and provides you with an integrated design experience. It is one of several project systems available for Visual Studio, and you can use many of the standard Visual Studio tools, editors, and other user interfaces. For example, you can use Solution Explorer to manage BizTalk projects within a business solution, and the Properties window to view and edit properties for files, projects, and solutions.

The content in this section assumes that you are familiar with Visual Studio 2005. If you are not familiar with Visual Studio 2005, review the Visual Studio 2005 documentation. The content in this section focuses on the BizTalk project system.

**In This Section**

- About BizTalk Project System
- Using the BizTalk Project System
- Using Visual Studio
- Working with BizTalk Projects

- Using BizTalk Type Browser
- About BizTalk Namespace References Included in BizTalk Projects

## About BizTalk Project System

You use the BizTalk® project system to create part or all of a Microsoft® BizTalk® Server application or business solution. The core element of any such solution is a BizTalk project—a collection of items, such as schemas, orchestrations, Web message types, classes, pipelines, maps, and references that you can build and generate into an assembly before deploying it.

A relatively simple business solution might consist of a single BizTalk project generated into a single assembly. If your business solution is more complex—for example, you have a solution that integrates many diverse systems and processes—a possible BizTalk solution might have many assemblies generated from many BizTalk projects and deployed to several BizTalk Server computers.

When you create a BizTalk project, you generally include one or more of the file types in the following list. These file types play specific roles in creating your solution, and the BizTalk project system provides you with a corresponding graphical design tool for each of them.

**Orchestrations.** An orchestration represents the workflow of a business process. You use the Orchestration Designer for designing orchestrations. For more information about Orchestration Designer, see *Creating Orchestrations Using Orchestration Designer*.

**Schemas.** A schema describes the structure of an XML document. Schemas exchange information among applications within an organization or among trading partners. BizTalk Editor simplifies the process of defining schemas. For more information about BizTalk Editor, see *Creating Schemas Using BizTalk Editor*.

**Maps.** A map transforms data from one format to another. BizTalk Mapper presents source schemas and destination schemas side-by-side and enables you to define transformations between data elements of different messages. For more information about BizTalk Mapper, see *Creating Maps Using BizTalk Mapper*.

**Pipelines.** A pipeline performs a variety of operations to prepare incoming or outgoing messages for further processing. Pipeline Designer enables you to implement such operations as encryption and decryption, compression, reformatting, and validation. For more information about Pipeline Designer, see *Creating Pipelines Using Pipeline Designer*.

You can also use BizTalk Type Browser to view the items and references in BizTalk projects. For more information about BizTalk Type Browser, see *Using BizTalk Type Browser*.

BizTalk projects can coexist with other projects in Visual Studio. As with all project systems in Visual Studio, BizTalk projects can include other files, such as ASP.NET pages, and can refer to other projects and assemblies that you have created. For more information about the BizTalk project template, see *BizTalk Server Project Templates*. For more information about creating BizTalk projects, see *Creating BizTalk Projects*.

## Using the BizTalk Project System

You can use the BizTalk® project system to create, organize, and configure BizTalk solutions in the Microsoft® Visual Studio® 2005 environment. The topics and procedures in this section describe how to perform various tasks by using the BizTalk project system.

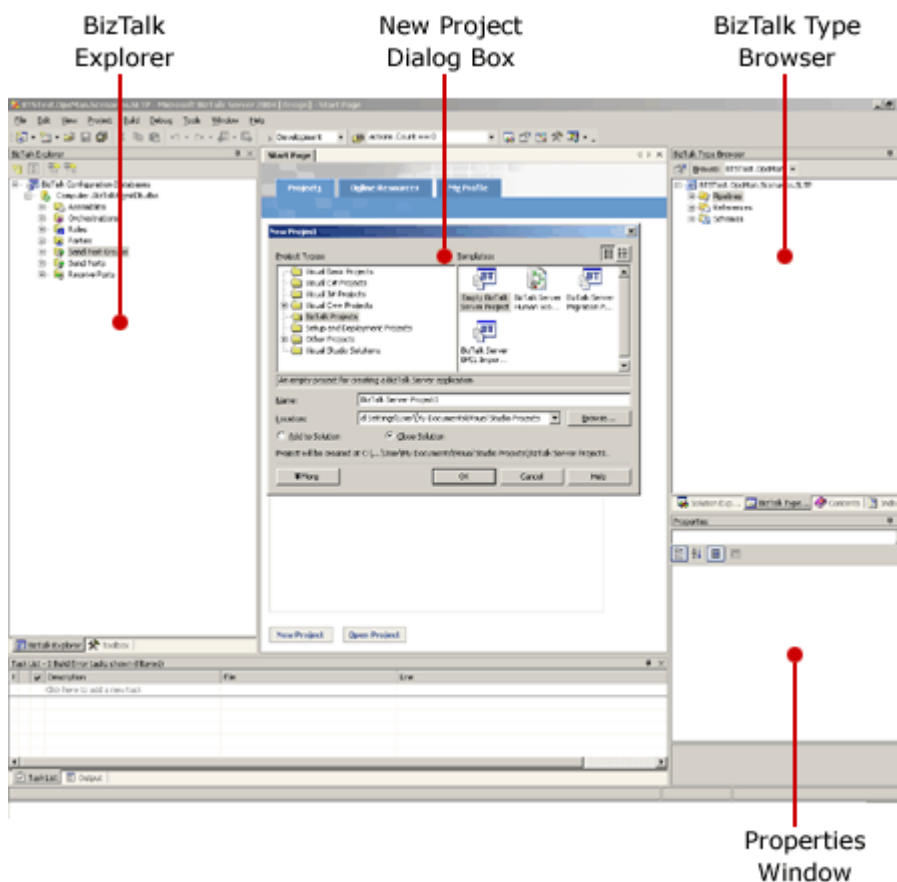
The BizTalk project system uses many of the same project management principles and procedures that you use with other project systems in Visual Studio. This section details common procedures that you might use when creating an application that runs on Microsoft BizTalk Server.

This section does not contain information or procedures that Visual Studio provides. For more information on creating applications in Visual Studio, see "Designing Distributed Applications" in the Visual Studio Combined Collection at <http://go.microsoft.com/fwlink/?LinkId=25488>.

For more information about using the Visual Studio environment, see "Managing Solutions, Projects, and Files" in the Visual Studio Combined Collection at <http://go.microsoft.com/fwlink/?LinkId=25486>.

The following figure shows the BizTalk project system design environment with the **New Project** dialog box open.

### Depicts the Project System Design Environment



**To open BizTalk Editor**

In Solution Explorer, click a schema.

On the **View** menu, click **Open**.

—Or—

Right-click the schema, and then click **Open**.

—Or—

Double-click the schema.

The selected schema opens in BizTalk Editor.

**To open BizTalk Mapper**

In Solution Explorer, click a map.

On the **View** menu, click **Open**.

—Or—

Right-click the map, and then click **Open**.

—Or—

Double-click a map.

The selected map opens in BizTalk Mapper.

**To open Orchestration Designer**

In Solution Explorer, click an orchestration (.odx).

On the **View** menu, click **Open**.

—Or—

Right-click the orchestration, and then click **Open**.

—Or—

Double-click the orchestration.

Orchestration Designer opens.



## To open Pipeline Designer

In Solution Explorer, click a pipeline.

On the **View** menu, click **Open**.

—Or—

Right-click the pipeline, and then click **Open**.

—Or—

Double-click the pipeline.

Pipeline Designer opens.

## Using Visual Studio

Within the BizTalk® project system, you can use many of the tools available in Microsoft® Visual Studio® 2005, as well as tools designed specifically for creating applications that run on Microsoft BizTalk® Server 2006. This section describes some of the common procedures that you can use to create an application that runs on BizTalk Server.

When you use the BizTalk project system, you use many of the same user interface (UI) components, such as the Solution Explorer and the Properties window, to create your application. Additionally, there are components, such as BizTalk Type Browser and BizTalk Editor that are available only after you install BizTalk Server 2006. While you can use these specific BizTalk UI components with any project system, they specifically enable you to build applications that run on BizTalk Server.

This section describes the various user interface components that you might use when you work with the BizTalk project system.

While the BizTalk project system uses many of the same menus and menu commands as other Visual Studio project systems, some commands are new, unavailable, expanded, or restricted when you use the BizTalk project system. This section describes the various menus available in Visual Studio and how they interact with the BizTalk project system.

### In This Section

- Using the File Menu
- Using the View Menu
- Using the Project Menu
- Using the Build Menu
- Using the Debug Menu

- Using the BizTalk Menu
- Using the Help Menu
- Using Property Pages

## Using the File Menu

Most of the **File** menu commands work the same way for BizTalk projects as for other Visual Studio projects. Certain commands are unsupported or unavailable when working with the BizTalk projects. For example, the **Print** command is not supported when you are working with pipelines.

## Using the View Menu

The following table lists BizTalk project system windows, toolbars, and toolboxes available from the **View** menu.

Submenu name	Submenu name (if applicable)	Description
<b>BizTalk Explorer</b>		BizTalk Explorer is an available window that provides a comprehensive view into the BizTalk Management database. You can also configure ports in BizTalk Explorer. For more information about BizTalk Explorer, see Using BizTalk Explorer.
<b>Other Windows</b>	<b>BizTalk Type Browser</b>	BizTalk Type Browser is an available window that enables you to view all of the items in a BizTalk project. The items are organized in a hierarchical fashion, enabling you to view the relationships between the different items within the context of your business solution. For more information about the BizTalk Type Browser, see Using BizTalk Type Browser.
<b>Other Windows</b>	<b>Orchestration View</b>	Orchestration View is an available window that enables you to add, delete, and examine orchestration parameters, ports, and port types, messages and multipart message types, correlation sets and correlation types, role links and role link types, scopes, and orchestration properties.
<b>Other Windows</b>	<b>Expression Editor</b>	Expression Editor is an available window that is a standard Visual Studio text editor with IntelliSense that enables you to enter complex expressions.
<b>Toolbox</b>	<b>BizTalk Pipeline Components</b>	This is a list of the pipeline components that you can drag onto the pipeline design surface. You can only add pipeline components to your active pipeline that are available.
<b>Toolbox</b>	<b>BizTalk</b>	This is a list of the orchestration shapes that you can drag onto the

	<b>Orchestrations</b>	orchestration design surface.
<b>Toolbox</b>	<b>BizTalk Mapper</b>	This is a list of the functoids that you can drag onto the map grid surface. The functoids are grouped by function.
<b>Toolbars</b>	<b>BizTalk Editor</b>	A visual tool that simplifies the process of creating structured document schemas, specified in XML Schema definition language (XSD), for both XML and non-XML formats.
<b>Toolbars</b>	<b>BizTalk Mapper</b>	A graphical user-interface tool that simplifies the process of specifying an XML document transformation, based on two schemas created with BizTalk Editor, producing an Extensible Stylesheets Language Transformations (XSLT) style sheet as compiled output.

## Using the Project Menu

The following table lists some of the commands on the **Project** menu.

Submenu name	Description
<b>Add Reference</b>	Use this menu item to reference other projects, other .NET projects, or COM projects.
<b>Add Web Reference</b>	Use this menu item to add Web references.
<b>Add Generated Items</b>	Use this menu item to add a generated adapter or schema file.

For information about adding Web references for BizTalk Web services, see [Adding Web References](#).

## Using the Build Menu

**Deploy Solution** on the **Build** menu deploys a BizTalk project. You should use this method of deployment only when you are developing your application or if you have a simple scenario. This method of deployment does *not* keep track of versions, and you can easily overwrite previous versions of an assembly. Reusing the same version is useful in a development or testing phase, but not in the production environment. For information on deployment, see [Understanding BizTalk Application Deployment and Management](#).

To add your BizTalk artifacts to the BizTalk Management database, run the Assembly Deployment Wizard. For more information about the Assembly Deployment Wizard, see [How to Deploy a BizTalk Assembly from Visual Studio](#).

## Using the Debug Menu

BizTalk project system does not support the **Debug** menu commands. BizTalk Server 2006 does not implement Visual Studio .NET debugging support. For information about alternatives for debugging in BizTalk Server 2006, see Debugging Orchestrations.

## Using the BizTalk Menu

When you work with a project, the **BizTalk** menu appears if you open BizTalk Editor, BizTalk Mapper, or Orchestration Designer. For more information about the **BizTalk** menu associated with BizTalk Editor, see Using BizTalk Editor Commands .

## Using the Help Menu

The following table lists some of the commands on the **Help** menu as they pertain to BizTalk Server 2006 Help.

Menu command	Description
<b>Dynamic Help</b>	This menu command opens the <b>Dynamic Help</b> tab that dynamically generates topics based on the task.
<b>Contents</b>	This menu command opens the <b>Contents</b> tab and displays all the installed Help collections. You must have the Microsoft Visual Studio product documentation and BizTalk Server product documentation installed to view the contents.
<b>About Microsoft BizTalk Server</b>	This menu command opens the <b>About Microsoft BizTalk Server</b> dialog box. This dialog box displays the BizTalk Server 2006 product information.
<b>Index</b>	The BizTalk Server 2006 Help documentation is not accessible through the index in this release.
<b>Search</b>	There is no filter for the BizTalk Server 2006 Help documentation in this release, but if you select <b>(no filter)</b> in the <b>Filtered by</b> drop-down list, the BizTalk Server 2006 Help documentation is available to searches.

## Using Property Pages

You use the **Property Pages** to configure the assembly project properties and deployment properties for your BizTalk project.

### To configure assembly project properties

Open the property pages for the project for which you want to configure assembly project properties.

In the **<ProjectName> Property Pages** dialog box, click the **Common Properties** node to display the common properties.

Select the **Assembly** node and update your assembly project properties.

### To configure deployment properties

Open the property pages for the project for which you want to configure deployment properties.

In the **<ProjectName> Property Pages** dialog box, click the **Configuration Properties** node to display the configuration properties.

Select the **Deployment** node and update your deployment properties.

## Working with BizTalk Projects

While many of the principles of creating an application with the BizTalk® project system are the same as those of creating applications with other project systems available in Microsoft® Visual Studio®, there are a few differences, such as the types of project build configurations. This section describes some of the tasks that you might perform when creating an application that runs on Microsoft BizTalk® Server.

### In This Section

- BizTalk Server Project Templates
- Creating BizTalk Projects
- Source Control and the BizTalk Project System

## BizTalk Server Project Templates

When you install BizTalk Server 2006, the installation process adds the BizTalk Projects folder to the **New Project** dialog box. The BizTalk Projects folder contains templates for creating an empty BizTalk Server Project, a BizTalk Server Human Workflow Project, a BizTalk Server Migration Project, and BizTalk Server BPEL Import Project.

The following table describes the BizTalk Project templates.

Use This	To Do This
<b>Empty BizTalk Server Project</b>	To create a new empty BizTalk Server project.
<b>BizTalk Server Migration Project</b>	To migrate BizTalk solutions to BizTalk Server 2006.
<b>BizTalk Server Human Workflow</b>	To create a new Human Workflow Services application.

Project	
<b>BizTalk Server BPEL Import Project</b>	To import Business Process Execution Language (BPEL), Web Services Description Language (WSDL), or XML Schema Definition Language (XSD) files into a BizTalk project.

## Creating BizTalk Projects

To create an application that runs on BizTalk Server, you start by adding one or more BizTalk projects to a solution. This section describes some of the tasks that you might perform when you work with BizTalk projects.

For more information about creating solutions and projects, see "Introduction to Solutions, Projects, and Items" in the Visual Studio .NET Combined Collection at <http://go.microsoft.com/fwlink/?LinkId=25492>.

### To create a BizTalk project

In Visual Studio .NET, on the **File** menu, point to **New**, and then click **Project**.

In the **New Project** dialog box, in the **Project Types** area, select **BizTalk Projects**.

In the **Templates** area, select one of the BizTalk Server Project templates.

Select **Close Solution** and click **OK**.

### To add a new BizTalk project to a solution

In Visual Studio .NET 2003, on the **File** menu, point to **New**, and then click **New Project**.

In the **Project Types** area, select **BizTalk Projects**, and in the **Templates** area, select one of the BizTalk Server Project templates.

Select **Add to Solution** and click **OK**.

### In This Section

- Adding Project Items
- Configuring Project Property Pages
- Configuration Properties
- Solution Build Configurations

## Considerations When Creating BizTalk Projects

This section provides information that you should take into consideration when creating BizTalk projects using Visual Studio 2005.

### **Avoid compilation errors caused by projects that are too large**

The Visual Studio compiler will not successfully compile a project if it would result in an assembly larger than 75 megabytes. When the compiler reaches a size constraint it will emit fatal error CS0013 "Unexpected error writing metadata to file <filename>" and halt.

To avoid this problem, we recommend that projects not exceed 10 megabytes unless absolutely necessary. Why?

Smaller projects are potentially simpler to deploy because there are fewer deployment steps. And with smaller projects the steps are more likely to be closely related -- managing trading partner discounts or handling RFP's.

It is easier to isolate bugs, deployment issues, and other problems when using smaller projects. Finding a bug in a project with 140 schemas and many custom maps and scripts will be more difficult than finding one in a project with only 10 schemas and a few custom maps and scripts.


Separating a large project into smaller projects can reduce complexity. The smaller projects are more manageable.

Smaller projects compile faster.

Splitting a large project with many unrelated schemas into smaller projects with strongly related schemas may result in better performance. This is because only some of the assemblies will be loaded at time.

### **Avoid using the Project Name as the Map Type Name**

When adding a new Map to a BizTalk project in Visual Studio 2005 do not use the project name as the type name. If you do, the compiler will generate one or more errors similar to "The type name <name>' does not exist in the type".

To change the type name for a map from within a BizTalk project, click on the map in the Solution Explorer pane, then verify the type name property in the Properties pane. If it is the same, you need to modify it making sure to change any configurations that rely on it.  Visual Studio Team System Is Not Directly Supported

BizTalk projects in Visual Studio 2005 do not directly support the Visual Studio Team System. Visual Source Safe is fully supported for the tracking and versioning of BizTalk project artifacts.

## Adding Project Items

In the context of the BizTalk project system, a project item is a configured item, such as a map or schema. A BizTalk application might contain one or more orchestrations, schemas, maps, and pipelines.

### Orchestrations

An orchestration is a representation of a business process expressed in the XLANG/s language. XLANG/s is an updated variant of the XML-based language (XLANG) introduced in Microsoft BizTalk Server 2000. XLANG/s can be used to complement existing procedural languages such as Microsoft® Visual C#® and Microsoft® Visual Basic® .NET.

You can use Orchestration Designer to create the orchestrations that you want to include in a BizTalk project. All orchestrations that you create in Orchestration Designer have an .odx file extension. For more information about orchestrations and the Orchestration Designer, see **Using Orchestration Designer**.

### Schemas

A schema is the definition of the structure for a document or message. It contains property information as it pertains to the records and fields within the structure. If appropriate, a schema can contain multiple subschemas.

You can use BizTalk Editor to import, edit, or create schemas. You can then use the schemas that you create to generate maps in BizTalk Mapper. All schemas that you save by using BizTalk Editor have an .xsd file extension.

For more information about schemas and BizTalk Editor, see *Creating Schemas Using BizTalk Editor*.

### Maps

A map is an XML file that defines the correspondence between the records and fields in one schema and the records and fields in another schema. You create maps based on industry standards, non-industry standards (such as internal standards or legacy issues), or existing files. You create maps when you want to transform data that you receive or send from one format to another. You can use BizTalk Mapper to create maps that you include in a BizTalk project. All maps that you save in BizTalk Mapper have a .btm file extension. For more information about BizTalk Mapper, see *Creating Maps Using BizTalk Mapper*.

### Pipelines

You can use Pipeline Designer to create receive and send pipelines. A pipeline is a software infrastructure that defines and links one or more stages for processing messages received or sent by BizTalk Server 2006. The pipelines implement the stages in a specific order and include functions such as encoding or decoding, assembling or disassembling, and encrypting or decrypting.



The default pipeline references included in a BizTalk project can process only XML documents. If you want to process flat files, EDI documents, or other file types, you must create new pipelines as appropriate. All pipelines created with Pipeline Designer have a .btp extension. For more information about pipelines and Pipeline Designer, see [Creating Pipelines Using Pipeline Designer](#) .

## Valid Files for BizTalk Projects

When you work with a BizTalk project, you can include many different files, such as HTML files, XML files, receive pipeline files, and schema files. However, when you build a BizTalk project into an assembly, only the following file types are included in the assembly: orchestrations, schemas, maps, and pipelines.

## Configuring Project Property Pages

Project properties configure global settings for projects that you create in Visual Studio. BizTalk projects share many of the project properties of other project systems. You configure these properties in the <***Name of Project*Property Pages** dialog box. For BizTalk projects, there are two types of global properties: common and configuration.

The Common Properties folder contains the following categories of properties:

**General.** These properties define general information about the project such as the name of the file, the default namespace, and the output type.

**Assembly.** These properties define information about the generated assembly such as the culture, version, and configuration.

**References Path.** These properties specify the directories to search for the references associated with the BizTalk project.

### In This Section

- Common Properties: General
- Common Properties: Assembly
- Common Properties: References Path

## Common Properties: General

In **General** properties, you configure the settings for the current project. This topic describes the properties that define the settings.

### Assembly Name

The **Assembly Name** property displays the name of the output file that contains the assembly metadata. If you change the value of this property, you also change the **Output Name** property value.

If you change the property name, the new name must follow naming conventions for Microsoft® Windows® files and cannot contain the following characters:

Dollar sign (\$)

At sign (@)

Angle brackets (< >), brackets ([ ]), braces ({ }), and parentheses (( ))

Colon (:) and semicolon (;)

Equal sign (=)

Caret sign (^)

Pipe (vertical bar) (|)

Asterisk (\*)

Exclamation point (!)

Forward slash (/) and backward slash (\)

Percent sign (%)

Question mark (?)

Comma (,)

Quotation mark (single or double) (' ")

Tab

## Output Type

The **Output Type** property specifies the type of application to build. This is a read-only property.

## Default Namespace

The **Default Namespace** property configures the default namespace for added items, such as artifacts, which are added by using the **Add New Item** dialog box. You can change the name of the default namespace by changing the value of the property.

## Project File

The **Project File** property displays the name of the project and contains build and configuration information about the project. This is a read-only property.

## Project Folder

The **Project Folder** property displays the directory of the project. For BizTalk projects, this is a read-only field.

## Output File

The **Output File** property displays the name of the compiled output of the project. This is also known as the name of the assembly, for example, <AssemblyName>.dll. This is a read-only property.

## Common Properties: Assembly

In **Assembly** properties you configure the settings for the assembly generated by the current project. All properties have read/write access unless otherwise specified. Additionally, all assembly properties map to the fields in the AssemblyInfo.cs file in a C# project. This topic describes the properties for the assembly settings.

For more information about assembly properties, see "Global Attributes" in the Visual Studio Combined Collection at <http://go.microsoft.com/fwlink/?LinkId=25503>.

### Assembly Culture

Use the **Assembly Culture** property to configure the culture that the assembly supports. Culture information can include language and calendar specifications.

### Assembly Version

Use the **Assembly Version** property to specify the version of the assembly. In BizTalk projects, the assembly version is composed of four parts and is in the following format: major.minor.build.revision.

### Assembly Company

Use the **Assembly Company** property to specify the name of the company.

### Assembly Copyright

Use the **Assembly Copyright** property to describe the copyright information for the assembly.

### Assembly File Version

Use the **Assembly File Version** property to configure the Win32® file version number. The value in this field defaults to the assembly version.

## Assembly Informational Version

Use the **Assembly Informational Version** property to specify the version information that appears in the **Assembly Properties** dialog box on the **Summary** tab. The runtime does not use this property. The value in this field defaults to the assembly version.

## Assembly Product

Use the **Assembly Product** property to specify the product information for the assembly. The value in this field defaults to the assembly version.

## Assembly Trademark

Use the **Assembly Trademark** field to describe the trademark information for the assembly. The value in this field defaults to the assembly version.

## Assembly Configuration

Use the **Assembly Configuration** property to specify the custom configuration of the assembly. For example, you could describe the assembly as "Retail," "Purchase Process," and so on. The runtime does not use data in this property.

## Assembly Default Alias

Use the **Assembly Default Alias** property to configure the default alias. The assemblies use this default alias to reference the project for which you are configuring the properties. This is a friendly name.

## Assembly Description

Use the **Assembly Description** property to describe the nature and purpose of the assembly.

## Assembly Title

Use the **Assembly Title** property to specify the title of the assembly. This is a friendly name that can include spaces.

## Assembly Delay Sign

Use the **Assembly Delay Sign** property to configure whether delay signing is used. You use a Boolean value in this property.

## Assembly Key File

Use the **Assembly Key File** property to specify the name of the file that contains the public key or the public and private key for the assembly.

## Assembly Key Name

Use the **Assembly Key Name** property to indicate the key container that stores the key pair used to create the strong name.

## Common Properties: References Path

In the References Path property you specify the directories to search for references associated with the project when the project is loaded. This setting is specific to the selected project, computer, and user for which it is set.

For more information about reference paths, see "Reference Path, Common Properties, <Projectname> Property Pages Dialog Box" in the Visual Studio Combined Collection at <http://go.microsoft.com/fwlink/?LinkId=25504>.

## Configuration Properties

In the Configuration Properties folder, you specify the target configurations for the project properties. The Configuration Properties folder contains the following:

**Build.** Use these properties to define information such as the output path and warning levels for when you build the project.

**Deployment.** Use these properties to configure information such as the server to which to deploy projects. For information on how to deploy assemblies, see *How to Deploy a BizTalk Assembly from Visual Studio*.

### In This Section

- Configuration Properties: Build
- Configuration Properties: Deployment

## Configuration Properties: Build

In the **Build** properties, you configure the attributes for the project build output. This topic describes the properties that you set on this project properties page. All properties described in this section have read/write access unless otherwise indicated.

### BPEL Compliance

Use this property to configure whether to generate Business Process Execution Language (BPEL) compliance output.

### Warning Level

Use this property to specify the warning level to use when you build the project. The default value is **4 (level 3 plus informational warnings)**.

### Treat Warnings As Errors

Use this property to specify whether to treat warnings as errors when you build the project. The default value is **false**.

### Embed Tracking Information

Use this property to configure whether to embed tracking information in the assembly. The default value is **true**.

### Generate Debugging Information

Use this property to configure whether to generate debug symbols and c# files when the project is built. The default value is **true**.

### Output Path

Use this property to configure the output files for the project configuration. The default value for this property is `\<ProjectFolder>\bin\<ProjectType>` where the value of `<ProjectType>` is "Development" or "Deployment."

## Configuration Properties: Deployment

In the **Deployment** properties, you configure the attributes for the BizTalk project deployment. You must configure the **Server**, **BizTalk Management Database**, and **Administration Group** properties as a set for deployment to be successful. For example, servers installed with BizTalk Server belong to an Administration group, and each Administration group uses a single BizTalk Management database.

### Server

This is the name of the server where the Configuration repository (also known as the BizTalk Management or Configuration database) is located. You deploy the BizTalk project to this server if you configure the BizTalk project as "Deployment."

### BizTalk Management Database

You use this field to specify the name of the BizTalk Management database for the deployed assembly. This applies if you have configured the BizTalk project as a deployment project.

### Administration Group

You use the **Administration Group** property to configure the name of the administration group for the deployed assembly. This applies if you have configured the BizTalk project as a deployment project.

## Redeploy

You use the **Redeploy** property to determine whether to delete the existing configuration and to re-create the configuration each time you deploy the assembly. The default value is **No**.

## Install to Global Assembly Cache

You use the **Install to Global Assembly Cache** property to indicate if Visual Studio .NET needs to install BizTalk assembly to the global assembly cache (GAC).

## Solution Build Configurations

As with other projects that you build in Visual Studio, you can use Configuration Manager to specify solution build configurations. Solution build configurations enable you to determine which projects to include in different builds of a solution. For BizTalk projects, you can use two default solution build configurations: development or deployment.

A development solution build configuration enables you to build a solution and to generate an assembly when you are finished. When you create a BizTalk project, the default solution build configuration is development.

A deployment solution build configuration implements the same features and functionality as a development configuration, and it also enables you to deploy the assembly after you build it.

### To configure build properties in Configuration Manager

From the **Build** menu, select **Configuration Manager**.

In the **Configuration Manager** dialog box, select one of following to configure the build properties.

The following table describes the configuration settings.

Use this	To do this
<b>Deploy</b>	To change your deployment configuration.
<b>Development</b>	To change your development configuration.

## Source Control and the BizTalk Project System

Source control is a mechanism in which a central piece of software stores and tracks file versions and controls who can access the files. When you use source control, you can:

Limit the possibility of overwriting important files.

Apply version numbers to your files.

Archive older versions of a source-controlled file.

Keep track of who modified a file, when they modified it, and what they modified.

Visual Studio enables you to integrate with any Microsoft Source Code Control Interface (MSSCCI)-compliant system. The BizTalk project system supports all of the source control features that are available in Visual Studio.

## Using BizTalk Type Browser

BizTalk® Type Browser is a tool available in Microsoft® Visual Studio® .NET after you install Microsoft® BizTalk® Server 2006. You can use BizTalk Type Browser to view all BizTalk-specific objects, such as orchestrations, schemas, maps, message types, port types, and operations, in a BizTalk project. This tool is different from BizTalk Explorer and Solution Explorer. The BizTalk Type Browser displays all items associated with the selected project in a meaningful hierarchy in the context of the business solution that you are creating. For example, you can view all the schemas and maps that are included in or referenced by the selected BizTalk project. You can view the relationship of all the objects.

The following types of objects can appear in BizTalk Type Browser, depending on what you included in your BizTalk project:

Schema

Map

Pipeline

These items only appear in BizTalk Type Browser if they are included in the selected BizTalk project. For example, if the BizTalk project includes maps and schemas, but not pipelines, only maps and schemas appear in the browser.

If your BizTalk project includes references to other projects or assemblies, the items in the referenced BizTalk projects and assemblies appear in BizTalk Type Browser. For example, if the current BizTalk project contains a reference to an assembly that contains maps, schemas, and pipelines, all of these items will appear in BizTalk Type Browser.

You can use BizTalk Type Browser to view the logical hierarchy of all of the types in your BizTalk project, such as referenced assemblies, schemas, maps, and pipelines.

### To open BizTalk Type Browser

On the **View** menu, point to **Other Windows**, and then click **BizTalk Type Browser**.

—Or—

Press CTRL+SHIFT+J when you are in Visual Studio .NET



For information about how to create a project, see [Creating BizTalk Projects](#). For information about adding items to a project, see [Adding Project Items](#).

### To view properties

In BizTalk Type Browser, select the node or item whose properties you want to view.

On the toolbar, click the **Properties Window** button.

—Or—

Press F4.

### To change BizTalk projects

In BizTalk Type Browser, in the **Browse** list, select the project that you want to view.

## BizTalk Type Browser Options

BizTalk Type Browser includes two options that enable you to:

**View the properties of an artifact.** Click the artifact to view the properties for the selected artifact.

**View a different BizTalk project.** You view other projects in your solution or change the project that appears in BizTalk Type Browser by selecting a different project from the **Browse** list in BizTalk Type Browser.

## BizTalk Type Browser Nodes

A node in BizTalk Type Browser can be one of the following types:

Container

Reference

Type

An orchestration can only be viewed from assembly references.

### In This Section

Root and Container Nodes

Project/Assembly or Reference Nodes

Type Nodes

## Root and Container Nodes

Root and container nodes are the high-level nodes for organizing a project hierarchy. The root and container nodes can include other containers, or they can hold items.

The following table describes the root and container nodes in BizTalk Type Browser.

Node	Description	Parent node
<Project name>	This node displays the name of the current BizTalk project.	None. This is the top-level node.
References	Container for referenced projects and assemblies.	Root node (<Project name>).
Schemas	Container for all schemas.	Parent project or assembly. This could be the current project (<Project name>), or a referenced project or assembly.
Orchestrations	Container for orchestrations.	Parent project or assembly. This could be the current project (<Project name>), or a referenced project or assembly.
Ports	Container for ports.	Parent project or assembly. This could be the current project (<Project name>), or a referenced project or assembly.
Port Types	Container for port types.	Parent project or assembly. This could be the current project (<Project name>), or a referenced project or assembly.
Roles	Container for roles.	Parent project or assembly. This could be the current project (<Project name>), or a referenced project or assembly.
Role Links	Container for role links.	Parent project or assembly. This could be the current project (<Project name>), or a referenced project or assembly.
Role Link Types	Container for role link types.	Parent project or assembly. This could be the current project (<Project name>), or a referenced project or assembly.
Messages	Container for messages.	Parent project or assembly. This could be the current project (<Project name>), or a referenced project or assembly.
Maps	Container for maps.	Parent project or assembly. This could be the current project (<Project name>), or a referenced

		project or assembly.
Pipelines	Container for pipelines.	Parent project or assembly. This could be the current project (<Project name>), or a referenced project or assembly.

## Project/Assembly or Reference Nodes

BizTalk project/assembly or reference nodes contain items that are referenced in a project or references to message types.

The following table describes the project/assembly or reference nodes in BizTalk Type Browser.

Node	Description	Parent node
<Referenced project/assembly name>	Contains items that are in a referenced project or assembly.	The References container node.
<Message type reference name>	Contains a reference to a message type from within an operation.	The Operations container node.

## Type Nodes

Type nodes are definitions of items such as schemas, maps, role links, ports, and pipelines that are contained in your project or in other referenced assemblies or projects.

The following table describes the type nodes in BizTalk Type Browser.

Node	Description	Parent node
<Schema name>	The name of an XSD-based schema.	The Schemas container.
<Map name>	The name of an XSLT-based map.	The Maps container.
<Role Link name>	The name of a role link.	The Project container.
<Port name>	The name of a port.	The Port container.
<Pipeline name>	The name of a pipeline.	The Pipelines container.

## About BizTalk Namespace References Included in BizTalk Projects

When you add a new BizTalk® project, the following namespaces are included by default:

**Microsoft.BizTalk.DefaultPipelines**

**Microsoft.BizTalk.GlobalPropertySchemas**

**System**

**System.Xml**

You can also add new references and Web references to your project. For more information about adding references using the **Project** menu, see [Using the Project Menu](#) . For information about adding Web references, see [Adding Web References](#) .

### In This Section

- [Microsoft.BizTalk.DefaultPipelines Reference](#)
- [Microsoft.BizTalk.GlobalPropertySchemas Reference](#)
- [System Reference](#)
- [System.Xml Reference](#)

## Microsoft.BizTalk.DefaultPipelines Reference

The **Microsoft.BizTalk.DefaultPipelines** namespace contains the following pipelines:

XMLReceive

PassThruReceive

XMLTransmit

PassThruTransmit

A pipeline is a software component that defines and links one or more stages for processing messages received or sent by BizTalk Server 2006. The stages include functions such as encoding or decoding, disassembling or assembling, and decrypting or encrypting. These functions are implemented in a specific order. You can use Pipeline Designer to create receive and send pipelines.

The default pipeline references included in a BizTalk project can process all types of documents using the **PassThruReceive** and **PassThruTransmit** pipelines.

The following lists show the default components in the default pipelines. These lists also indicate the default order of the components in each pipeline. You can add and delete components if necessary.

The default components in the default XMLReceive pipeline are:

Decrypter

Decoder

Disassembler

Validator

Party Resolution

The default components in the default XMLTransmit pipeline are:

Assembler

Encoder

Encrypter

## Microsoft.BizTalk.GlobalPropertySchemas Reference

The **Microsoft.BizTalk.GlobalPropertySchemas** namespace contains property schemas for the properties that various BizTalk Server components use. This namespace contains system properties that the BizTalk engine uses, transport specific properties that each transport uses for handling the configuration, and properties for configuring pipeline components.

The following table shows the global property schemas included in the **Microsoft.BizTalk.GlobalPropertySchemas** namespace.

Property schema	Feature area and description
bts-btf2-properties.xsd	Property schema.
bt2-endpoints-header.xsd	Schemas that define the BizTalk Framework constructs. These schemas are specific to BizTalk Framework Assembler and Disassembler pipeline components.
bt2-envelope.xsd	
bt2-manifest-header.xsd	
bt2-process-header.xsd	
bt2-properties-header.xsd	

btf2-receipt-header.xsd	
btf2-services-header.xsd	
bts-appstream-properties.xsd	The BizTalk Message Queuing adapter uses these property schemas for internal purposes.
MSMQT-properties.xsd	
bts-system-properties.xsd	This is a system property schema. The BizTalk engine uses most properties in this schema. You can use some properties for message routing. For more information on the properties that you can use for message routing, see <b>Message Context Properties</b> .
bts-endpoint-properties.xsd	This is an internal property schema.
bts-mime-properties.xsd	These are property schemas for pipeline components: MIME, XML, Flat File, and BizTalk Framework Assembler and Disassembler pipeline components.
bts-xmlnorm-properties.xsd	
bts-legacy-properties.xsd	BizTalk uses this schema for upgrading BizTalk Server 2002 applications to BizTalk Server 2006 applications.
bts-messagetracking-properties.xsd	The tracking engine uses this schema.
bts-soap-properties.xsd	These are transport-specific property schemas. Transports use these schemas to carry specific transport information and configurations. For more information on transports, see Using Adapters .
bts-file-properties.xsd	
bts-http-properties.xsd	
bts-smtp-properties.xsd	
bts-ftp-properties.xsd	
XlangMozartProperties.xsd	The orchestration engine uses this schema.

## System Reference

The **System** reference namespace contains fundamental classes and base classes that define commonly used value and reference data types, events and event handlers, interfaces, attributes, and processing exceptions. This default reference is necessary for all BizTalk projects to function properly. Do not delete the **System** reference.

For more information about the **System** reference, see "System Namespace" in the Visual Studio Combined Collection at <http://go.microsoft.com/fwlink/?LinkId=25494>.

## System.Xml Reference

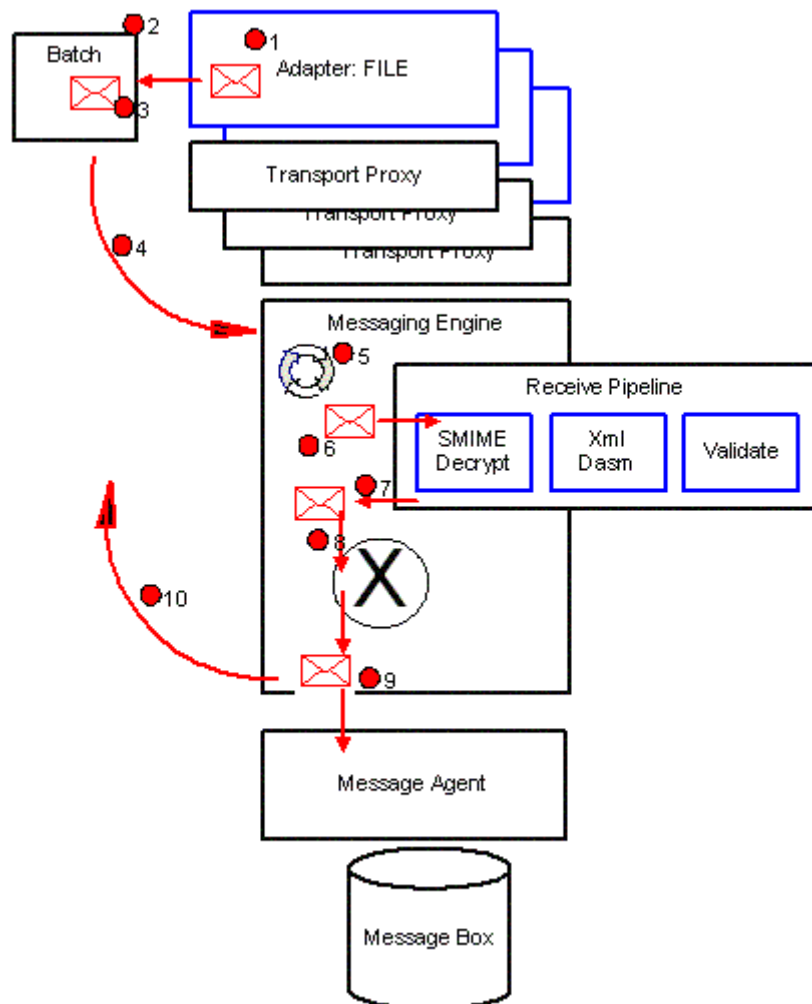
The **System.Xml** reference namespace provides standards-based support for processing XML documents. This default reference is necessary for all BizTalk projects to function properly. It is highly recommended that you do not delete this default reference.

For more information about the **System.Xml** reference, see "System.Xml Namespace" in the Visual Studio Combined Collection at <http://go.microsoft.com/fwlink/?LinkId=25498>.

## Using the BizTalk Messaging Engine

The following diagram illustrates the architecture of the Messaging Engine. It shows a scenario in which a message is received by an adapter and submitted into BizTalk Server.

### Architecture of the Messaging Engine



Each adapter has its own instance of a **TransportProxy** object that it uses to interact with the Messaging Engine. Adapters perform work against the Messaging Engine in batches, which are processed atomically. A batch is a collection of operations such as `SubmitMessage`, `SuspendMessage`, or `DeleteMessage`.

The following is the sequence of events for the scenario where an adapter submits a message to the Messaging Engine:

The adapter creates a new message and connects the data stream to the message.

The adapter gets a new batch from the Messaging Engine.

The adapter adds the message to the batch to be submitted.

The batch is committed and queued up on the Messaging Engine thread pool.

The Messaging Engine thread pool starts processing the new batch.

The message is processed in the receive pipeline.

The receive pipeline produces zero or more messages. Pipelines can consume messages providing they do not return any errors. Receive pipelines can produce more than one message; typically this happens when the disassembler component disassembles a single interchange into many messages. Typically the receive pipeline normalizes the submitted message into XML.

The message(s) produced by the pipeline will be processed in the mapper if mapping is configured.

The message(s) are published to the Message Agent or to the MessageBox database.

The Messaging Engine calls back the adapter to notify it of the outcome of the batch of work.

### **In This Section**

- Interchange Processing
- Ordered Delivery of Messages
- Error Handling

## **Interchange Processing**

This section discusses interchange processing at the disassembly stage of a receive pipeline. It examines interchange processing modes and different failure points within the messaging pathway.

### **Processing Modes**

Interchanges are processed at the receive pipeline disassembly stage in two modes:



- **Standard mode.** When the disassembler component of a receive pipeline is configured to perform standard disassembly, the messages contained in an interchange are extracted, processed by the receive pipeline, mapped, and published as a transactional unit of work. That is, either the entire interchange and its contained messages are fully processed and published into the MessageBox database, or the interchange is placed in the Suspended queue.
- **Recoverable mode.** When the disassembler component of a receive pipeline is configured to perform recoverable interchange processing, the messages contained in an interchange are extracted independently of each other. Messages that are successfully extracted are propagated further down the receive pipeline. Messages that are identified within an interchange but are not successfully extracted are placed in the Suspended queue.

After the messages within an interchange are identified and are either propagated further down the pipeline or placed in the Suspended queue, the subsequent processing of those messages is again treated transactionally. If any message fails at any point along the messaging pathway, except in its routing (for example, no matching subscriber), then all of the messages are discarded and the interchange from which they originated is placed on the Suspended queue.

Recoverable interchange processing is motivated by the need to support successful processing of a single interchange that contains multiple identifiable messages so that valid messages are propagated through the messaging pathway and invalid messages are suspended. With standard interchange processing, the existence of any invalid message in a given interchange causes the entire interchange to be suspended even if it contains one or more valid messages.

## Relevant Messaging Pathway Failure Points

Failures in the following specific areas of the messaging pathway result in the entire interchange being suspended:

Validation

Party Resolution

Mapping

## Examples

The following examples illustrate interchange processing scenarios.

### Example 1

In this example, the following pseudo-interchange is submitted on a standard interchange receive location. That is, the disassembler component in the receive pipeline is configured for standard interchange processing.

This interchange contains five messages, all of which the disassembler successfully extracts from the interchange. They are processed as follows:

The first, second, and third messages propagate through the pipeline and are ready to be published.

The fourth message fails processing at the disassembling stage in the pipeline. This causes all the messages that have already been processed to roll back and the original interchange message to be suspended as resumable.

The result of submission is:

Nothing is published. The original interchange is suspended because in standard interchange processing, any extracted message that fails at any point during or after interchange processing results in all extracted messages being discarded and the original interchange being placed on the Suspended queue as resumable.

## Example 2

The example uses the same interchange processing and propagation scenario as the previous example, except that the disassembly stage is configured to do recoverable interchange processing.

The result of submission is that individual extracted messages are all processed and the original interchange is discarded. The individual messages are processed as follows:

The first, second, and third messages propagate through the pipeline and are ready to be published.

The fourth message fails disassembly processing (that is, something goes wrong after it is extracted) and is ready to be suspended.

The fifth message propagates through the pipeline and is ready to be published.

After all messages are extracted from the interchange, messages 1, 2, 3, and 5 are published into the MessageBox database, and message 4 is placed on the Suspended queue. Message 2 is also redirected to the Suspended queue because of a routing failure due to no matching subscriber.

## Configuring Recoverable Interchange Processing

Recoverable interchange processing is a property of the disassembler component of a receive pipeline. Not all disassembler components allow you to specify recoverable interchange processing; for example, the native BizTalk Framework disassembler does not. If a disassembler supports recoverable interchange processing, then you specify this behavior in Pipeline Designer within Visual Studio 2005 when you add the disassembler component to the Disassemble stage of the pipeline being designed. After you drag the selected disassembler onto the Disassemble stage of the pipeline, you set the recoverable interchange processing property of that disassembler component to **true**.

## Party Resolution

Messages that are successfully extracted in recoverable interchange processing have their sending party identified according to the party configured for the receive port on which the parent interchange arrived. If party resolution fails for any message extracted from a given interchange, then party resolution is considered to have failed for the entire interchange.

## Restrictions

Recoverable interchange processing is not supported for interchanges that arrive through a receive location that uses the MSMQT adapter. This is not enforced at design time, but results in a run-time error in the pipeline processing.

When an interchange fails in the disassembler, the interchange is suspended as resumable. However, if the interchange is administratively resumed, it will fail again because the kinds of errors that cause disassembly failure are solely a result of the interchange content, which stays the same when the interchange is resumed. Such a failed interchange must be modified and resubmitted through a receive location.

If a message that was successfully extracted from an interchange subsequently fails to propagate through messaging due to an unmatched subscriber, the message is suspended as resumable. This message can be administratively resumed if the cause of the routing failure is fixed.

The disassembler component continues to extract messages from an interchange despite the following errors in disassembler components:

- Schema not found

- Schema ambiguity not resolved (that is, more than one schema exists for the same message type)

- XML validation failed

- Flat-file parsing failed

The disassembler component does **not** continue to extract messages from an interchange if the following error occurs in disassembler components:

- Document data is not well-formed XML—any document properties that would cause System.Xml.XmlReader to fail

## Ordered Delivery of Messages

Ordered message delivery ensures that messages that are published to the MessageBox database in a given order are delivered to each matching subscriber in the same order in which they were published to the message box.

## Configuring Ordered Message Delivery

Ordered message delivery can be configured in the following places:

**Receive** shape in an orchestration

Send port

## Ordered Delivery with Existing Transports

The protocols underlying certain transports, such as FILE and HTTP, are not consistent with the notion of ordered delivery. However, even with such transports, if the port bound to the transport is marked for ordered delivery, then BizTalk Server enforces ordered delivery by ensuring that the transport does not get the next outbound message until the current one has been successfully sent. To achieve this, BizTalk Server passes each message to the transport's adapter in a single batch and waits until the adapter has successfully deleted the message from the message box before delivering the next message, in another batch, to the adapter.

## Ordered Delivery for Custom Adapters

There are special considerations for custom receive adapters. If you are writing a custom adapter that supports ordered delivery on receive, the adapter should do the following:

After submitting a batch of messages, your custom receive adapter should wait for the **BatchComplete** callback from BizTalk Server before submitting the next batch. For more details, see [Batch-Supported Receive Adapter Creation](#) .

If a message fails in the pipeline, it should be suspended, preferably as nonresumable. Use the **BTS.SuspendAsNonResumable** message context property in BizTalk Server 2006 to flag the message appropriately.

For more details on building a custom adapter, see [Creating Custom Adapters](#) .

## Conditions for End-to-End Ordered Message Processing

To provide end-to-end ordered delivery the following conditions must be met:

Messages must be received with an adapter that preserves the order of the messages when submitting them to BizTalk Server. In BizTalk Server 2006, examples of such adapters are MSMQ, MQSeries, and MSMQT. In addition, HTTP or SOAP adapters can be used to submit messages in order, but in that case the HTTP or SOAP client needs to enforce the order by submitting messages one at a time.

You must subscribe to these messages with a send port that has the **Ordered Delivery** option set to **True**.

If an orchestration is used to process the messages, only a single instance of the orchestration should be used, the orchestration should be configured to use a sequential convoy, and the **Ordered Delivery** property of the orchestration's receive port should be set to **True**.

## Restrictions

Ordered delivery of messages is not supported for the following:

Dynamic send ports

Backup transports

## Interactions

When ordered delivery is configured for a send port, be aware of the following interactions with other configured behaviors in BizTalk Server:

For the "Stop sending subsequent messages on current message failure" setting on the same send port:

**False.** Only the failed message is suspended (as not resumable) and all subsequent messages continue to be processed. This preserves the ordering of the non-failed messages but can result in gaps in the sequence. For example, if orders 101, 102, and 103 are received and order 102 is suspended, orders 101 and 103 will continue to be processed in order.

**True.** The send port instance is suspended if any of the messages fails processing. This causes all subsequent messages in the ordered set of messages to be suspended. This preserves message order by preventing delivery of subsequent messages before delivery of the failed message.

If a solicit-response send port has the "Stop sending subsequent messages on current message failure" property set to **true**, and if recoverable interchange processing is configured for the disassembly stage of the receive pipeline for the response, then the send port does not stop sending messages (that is, the instance is not suspended) if there is a recoverable error in disassembling the response.

Before deleting an ordered send port, ensure that there are no instances associated with it. If there are associated instances, you should terminate them before deleting the send port.

## Ordered Delivery to File Transport

Messages arrive at the File adapter in sequence. The File adapter may append messages to a single file or write out a sequence of files, with the following results: .

If message data is appended to a single file, individual messages are appended in order.

If messages are written to individual files, the order is reflected in the file names, which are sequentially named. In this case, for files written by the adapter, file system properties relating to chronology (for example, file creation time or modification time) do not necessarily reflect the message arrival sequence.

## Performance Impact of Ordered Delivery

To achieve ordered delivery, BizTalk Server must serialize processing of ordered messages at various points along the message pathway. This works against scale-out techniques, such as the use of multiple host instances for parallel processing of messages. When using ordered delivery, even ports configured to run on multiple host instances run only on a single host instance to ensure ordered delivery. However, in this situation, high availability is still maintained because the failure

of a host instance that is processing ordered delivery of messages results in reprocessing of the failed message on another available host instance.

## Error Handling

The pathway that a message follows through the BizTalk Server messaging subsystem encompasses several points of processing and transfer. At each point along this pathway, failures can occur in both BizTalk Server infrastructure and application-provided elements such as custom pipeline components and orchestrations.

This section and those contained within discuss the typical failure modes of well-known stages of processing and how those failure modes are addressed through both BizTalk Server configuration and application-provided elements such as orchestrations. Failure effects include disposition of the message, diagnostics captured and logged, and operational considerations.

Many of these enumerated failure modes correspond to specific features of BizTalk Server that are designed to address the failure mode, such as recoverable interchange processing and error reporting. Other failure modes concern how BizTalk Server communicates failure information to application elements and, in turn, how application elements such as custom pipeline components, adapters, and orchestrations communicate failure information to BizTalk Server.

### In This Section

- Using Failed Message Routing
- Using Acknowledgments

## Using Failed Message Routing

The error-handling facility, which is new in BizTalk Server 2006, allows the designer to designate automated handling of messaging failures as an alternative to the traditional (now default) behavior of placing failed messages in the Suspended queue. This automated handling routes an error message to any subscribing routing destination, such as a send port or orchestration. The error message is a clone of the original message with all previously promoted properties now demoted and with selected properties related to the specific messaging failure promoted to the message context.

### What Does Failed Message Routing Consist Of?

When failed message routing is enabled, BizTalk Server does not suspend the message—it routes the message instead. Failed message routing can be enabled on both receive and send ports, with the following results:

If failed message routing is enabled on a receive port and a message fails in the receive pipeline or in routing, a failed message is generated. In the case where an error occurs in or before the disassembly phase, the error message is a clone of the original interchange.

If failed message routing is enabled on a send port and the message fails in the send pipeline, a failed message is generated.

When a failed message is generated, BizTalk Server promotes error-report-related message context properties and demotes regular message context properties before publishing the failed message. Compare this to the default behavior when failed message routing is not enabled: Messages that fail are suspended.

## What Kinds of Messaging Failures Trigger an Error Message?

Any failure that occurs in adapter processing, pipeline processing, mapping, or message routing results in an error message if routing for failed messages is enabled. When a messaging error occurs while an orchestration is receiving from a receive port or sending to a send port, the resulting error message is associated with the messaging ports to which the orchestration is bound.

## Subscribing to an Error Message

Error messages are delivered to orchestrations or send ports that have subscribed to receive them. A subscription typically selects an error message based on the name of the port in which the messaging error occurred (either a send or a receive port). A subscription might also filter on other properties promoted to the error's message context (for example, **InboundTransportLocation** or **FailureCode**).

## Error Message Specification

An error message is a clone of the original failed message, with all previously promoted properties demoted and with a set of error-specific properties promoted to the message context. Previously promoted properties are demoted to avoid unintended delivery to subscribers not designated to receive the error message. The error message is published for distribution to subscribers (orchestrations, send ports, and send port groups).

The properties that are promoted to the context of an error message all fall under the **ErrorReport** namespace. They are as follows:

Property name	Data type	Promoted	Description
FailureCode	System.String	Yes	Error code. A hexadecimal value that is reported in the BizTalk Server Administration console.
FailureCategory	System.Int32	Yes	This property is not used in BizTalk Server 2006. Its value is undefined.
Description	System.String	No	Error description. Same diagnostic text as is written to the Application Event Log regarding this messaging failure.
MessageType	System.String	Yes	Message type of failed message, or empty if message type is indeterminate.  BizTalk Server uses the message type to associate messages with their XML schemas. Message type is formed by

			concatenating the schema namespace with the schema root node: http://mynamespace#rootnode.
ReceivePortName	System.String	<p><b>Promoted</b> if the failure happened during inbound processing (in a receive port)</p> <p><b>Not promoted</b> if the failure happened in a send port.</p>	Name of the receive port where the failure happened.
InboundTransportLocation	System.String	<p><b>Promoted</b> if the failure happened during inbound processing (in a receive port)</p> <p><b>Not promoted</b> if the failure happened in a send port.</p>	URI of the receive location where the failure happened.
SendPortName	System.String	<p><b>Promoted</b> if the failure happened during outbound processing (in a send port)</p> <p><b>Not promoted</b> if the failure happened in a receive port.</p>	Name of the send port where the failure happened.



OutboundTransportLocation	System.String	<p><b>Promoted</b> if the failure happened during outbound processing (in a send port)</p> <p><b>Not promoted</b> if the failure happened in a receive port.</p>	URI of the send location where the failure happened.
ErrorType	System.String	Yes	Indicates the type of message that the error contains. In BizTalk Server 2006 this property always contains the value <b>FailedMessage</b> , meaning that the error contains the original failed message.
RoutingFailureReportID	System.String	Yes	This property provides the ID of the routing failure report that BizTalk Server generates when there is a routing failure. A routing failure report is a special message that BizTalk Server generates and suspends. This message does not have a body, but it has the context of the failed message. Using this ID, an error-handling orchestration or a send port can query the MessageBox database and process the routing failure report. For example, an orchestration may want to terminate the routing failure report after it gets the failed message.

## Enabling Routing For Failed Messages

Failed message routing is a property of send and receive ports, and is enabled by indicating "Enable routing for failed messages" on the port's property page.

To configure failed message routing for a receive port, perform the following steps (this applies to both one-way and request-response receive ports):

Open the BizTalk Server Administration console.

Expand **BizTalk Server 2006 Administration**, expand **BizTalk Group**, expand **Applications**, and then expand the application to which the receive port belongs.

Click the **Receive Ports** folder

In the right pane, double-click the name of the receive port you want to configure.

On the receive port's property page, in the left pane, select the **General** category.

In the right pane, select the **Enable routing for failed messages** check box.

To configure failed message routing for a send port, perform the following steps (this applies only to static one-way and static solicit-response send ports):

Open the BizTalk Server Administration console.

Expand **BizTalk Server 2006 Administration**, expand **BizTalk Group**, expand **Applications**, and then expand the application to which the send port belongs.

Click the **Send Ports** folder.

In the right pane, double-click the name of the send port you want to configure.

On the send port's property page, in the left pane, select the **Transport Advanced Options** category.

In the right pane, in the **Transport Options** group box, select the **Enable routing for failed messages** check box.

## Handling Error Messages

Error handling is specified by an orchestration or send-port subscription whose filter matches the properties that have been promoted to the message context of the error message.

## Security Implications

The identity associated with the original message—either its initial identity or its final identity determined by the Resolve Party stage of the receive pipeline—is assigned to the error message.

The security mechanisms that restrict delivery of messages to authorized subscribing ports and orchestrations also apply to error messages.

A send port that subscribes to an error message, but is not configured with an appropriate decryption certificate, does not receive error messages that result from messaging failures at or before the decrypt stage of the receive pipeline through which the original message entered BizTalk Server. Instead, the failed messages are placed in the Suspended queue.

## Adapter Messaging Failure

If an adapter suspends a message, an error message is published. No error message is generated if the message is not suspended.

## Transactional Receive Pipelines

If a transactional receive pipeline throws an exception (specifies that the transaction should be aborted), then the transaction is aborted and an error message is published.

If a transactional receive pipeline explicitly suspends a message (specifies that MessageDestination = SuspendQueue), then the current transaction is allowed to proceed (and may be committed unless subsequent stages specify to abort it) and the resulting error message is published.

## Solicit-Response Send Ports

When a request message is sent from an orchestration and it fails transmission or its response fails inbound processing, the orchestration gets an exception, regardless of whether the failed message has been routed.

In the case where a solicit-response send port is connected to a request-response receive port, the receive port gets either a response message (if the transmission succeeds) or a NACK (if the transmission fails), regardless of whether the failed message has been routed.

## One-Way Send Ports

When a message is sent from an orchestration through a send port configured for delivery notification, then the orchestration receives a delivery notification regardless of whether the error message has been routed. In other words, the send port generates a delivery notification for the orchestration even if the port encounters a messaging failure during processing. The notification confirms delivery to the port, but does not address successful processing through the port.

## Resuming Suspended Messages

Most messages that fail inbound processing (that is, processing from and including the receive adapter and up to but not including publication to the message box), and whose failures are not handled, are suspended as resumable. The exception is that request messages from two-way receive ports are suspended as nonresumable.

Messages are typically suspended in their original form (as they were before pipeline processing), with two exceptions:

**Messages suspended by pipeline components.** BizTalk Server suspends this type of message in the same form as it was provided to the failing pipeline component. When the message is resumed, it undergoes pipeline processing from the beginning of the same pipeline. This implies that a pipeline component in a pipeline stage that precedes the stage where the original failure occurred must be prepared to handle the "same" message in a form that is different from the original form in which it processed that message.

**Messages from recoverable interchange disassembly that subsequently fail routing.** BizTalk Server suspends this type of message in the same form as it was published. This is the form the message had **after** pipeline execution. When the message is resumed, it skips pipeline processing and is published directly to the MessageBox database.

## Scenarios Leading to Suspended (Non-Resumable) Messages

While it is more common for messages to be suspended as resumable, there are some scenarios that lead to non-resumable messages:

In an Ordered Delivery send port with continue on failure enabled, if there is a failure in the pipeline, mapping or transmission.

In an Ordered Delivery receive port, if the adapter is configured to suspend messages on non-resumable on failure. For example, if the MSMQ adapter setting "On Failure" is set to "Suspend (non-resumable)" or the MQSeries adapter has "Suspend as Non Resumable" enabled, failed messages will be suspended as non-resumable.

In a two-way receive port, if the response message fails in the pipeline, mapping, or transmission.

In a two-way receive port, if the receive message fails in the pipeline, mapping or transmission. Individual adapter behavior may be different. For example, the HTTP adapter does not suspend messages by default but can be configured to do so.

## Using Acknowledgments

The BizTalk Messaging Engine generates positive acknowledgments (ACK) and negative acknowledgments (NACK) in response to conditions encountered during the processing of a message through a port. BizTalk Server publishes a positive acknowledgment to indicate successful transmission of a message and a negative acknowledgment to indicate transmission failure and suspension of a message.

### Why Use Acknowledgments?

Positive and negative acknowledgments provide strong feedback that you can use to determine if a message arrived at its destination or encountered one or more problems along the way. For example, acknowledgments are useful when:

You want to monitor a receive port for a new trading partner for schema validation and other errors.

You want to mark the status of a loan request sent out for approval as "in process" if it is successfully sent to a partner for approval or "failed" if transmission fails (for example, if the partner's server is down).

You process interchanges containing multiple purchase orders and want to track the number of orders that are transmitted or fail transmission.

You can accomplish each of these scenarios by using acknowledgments and content-based routing that uses filters.

### Routing Acknowledgments

When an ACK or NACK is published, all of the message context properties from the message that caused the ACK/NACK are demoted. Any properties that were promoted do not flow to the

acknowledgment. To route an acknowledgment, build a filter using the following properties from the **BTS** namespace:

Property name	Data type	Description
BTS.AckFailureCategory	xs:int	Identifies the <b>ErrorCategory</b> , which gives the place and reason for the suspension.
BTS.AckFailureCode	xs:string	Identifies the <b>ErrorCode</b> , which gives the place and reason for the suspension.
BTS.AckType	xs:string	Value is ACK for a positive acknowledgment and NACK for a negative acknowledgment.
BTS.AckID	xs:string	Identifies the <b>MessageID</b> of the original message.
BTS.AckOwnerID	xs:string	Identifies the instance ID from the original message.
BTS.CorrelationToken	xs:string	Identifies the correlation token from the original message if one is present.
BTS.AckDescription	xs:string	Identifies the <b>ErrorDescription</b> , which gives the place and reason for the suspension.
BTS.AckSendPortID	xs:string	Identifies the <b>SendPortID</b> from the original message.
BTS.AckSendPortName	xs:string	Identifies the <b>SendPortName</b> from the original message.
BTS.AckOutboundTransportLocation	xs:string	Identifies the <b>OutboundTransportLocation</b> from the original message.
BTS.AckReceivePortID	xs:string	Identifies the <b>ReceivePortID</b> from the original message.
BTS.AckReceivePortName	xs:string	Identifies the <b>ReceivePortName</b> from the original message.
BTS.AckInboundTransportLocation	xs:string	Identifies the <b>InboundTransportLocation</b> from the original message.

### Negative Acknowledgment Message Body

Much of the important information about a positive or negative acknowledgment is contained in the context properties. In addition to the context properties, NACKs also contain a message body part containing a SOAP fault. The format of the SOAP fault is as follows:

The exception message raised by the adapter is in the SOAP Detail section in the ErrorDescription element.

### **Accessing the Message Body from an Orchestration**

If you have an orchestration port that requires delivery notification, the `DeliveryFailureException` that is thrown on a transmission failure is deserialized from the SOAP fault that is contained in the NACK message body. To access the exception message string from within your orchestration, cast the `DeliveryFailureException` to a `SoapException` and then access the `InnerXml` as shown in the following code:

The preceding code sample returns an XML fragment similar to the following:

### **When Is an Acknowledgment Published?**

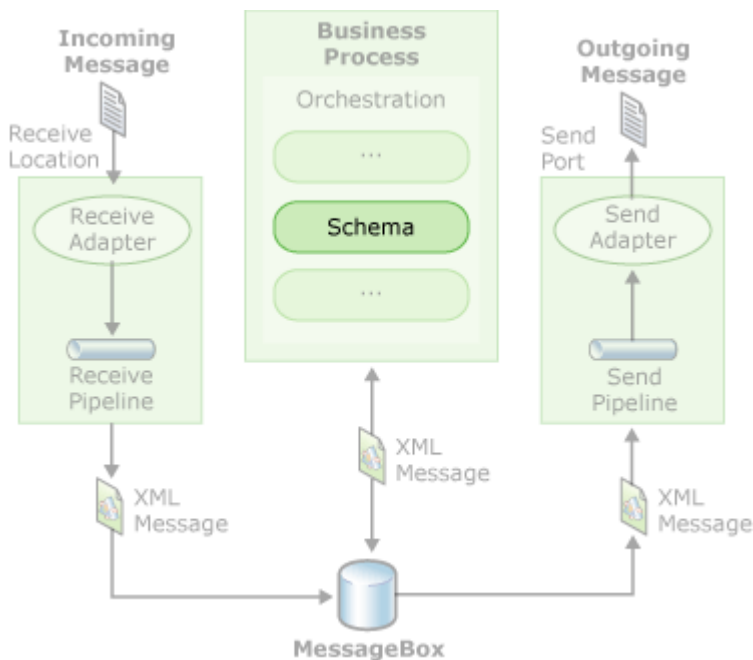
Both positive (ACK) and negative (NACK) acknowledgments are published to the MessageBox database at the point of failure provided there is at least one matching subscription. For example, if BizTalk Server cannot find a matching schema for a message read from a receive port and there are no NACK subscriptions, it suspends the message (if failed message routing has not been enabled) and does not publish a NACK.

## **Creating Schemas Using BizTalk Editor**

BizTalk® Editor is a tool that runs within the Microsoft® Visual Studio® 2005 environment. You can use it to create, edit, and manage schemas for use with your application. BizTalk Editor uses its own graphical system of hierarchical records and fields to represent the structure of instance messages, and uses the XML Schema definition (XSD) language to store the schemas that it defines. This is true regardless of the format in which instance messages are exchanged. For example, suppose that you exchange flat files with a trading partner. As BizTalk Server processes those flat files, it converts them to and from an XML format that conforms to an XSD schema that you defined in BizTalk Editor.

The schemas you create using BizTalk Editor can be used within an orchestrated business process, as shown in the following figure.

## Schemas in Business Processing



Schemas are also used by assemblers and disassemblers for translating instance messages from one format to another, such as between a flat file format and XML. Schemas also play an important role in instance message transformation, wherein the data in an instance message is used to construct an instance message with a different structure. The new instance message might be semantically equivalent, such as different representations of a purchase order, or it might be a different but related type of instance message that requires some or all of the data from the original instance message in its content.

An important reason for translating all instance messages into an XML format that conforms to an XSD schema is to simplify the process of transforming a message from one structure into another structure. Message structures are typically semantically equivalent despite their syntactic differences. For example, you and your trading partner might structure your purchase orders differently, but the basic information they contain is the same, allowing them to be transformed back and forth automatically. By first converting all instance messages into an XML format governed by a corresponding XSD schema, the instance messages can be translated back and forth between XML and non-XML formats, and transformed from one XML structure to another. For more information about the distinction between instance message translation and instance message transformation, see [Transformation vs. Translation](#).

The companion tool to BizTalk Editor within the Microsoft Visual Studio .2005 environment is BizTalk Mapper. After you use BizTalk Editor to create the schemas that define the structure and format of a pair of related instance messages, you use BizTalk Mapper to graphically define how to transform an instance message conforming to one schema (the source instance message and schema) into an instance message conforming to another schema (the destination instance message and schema). The specification of such transformations is implemented using Extensible Stylesheet Language Transformations (XSLT) and persisted as files called maps. For conceptual and procedural information about BizTalk Mapper, see [Creating Maps Using BizTalk Mapper](#). For reference information about BizTalk Mapper properties and functoids, see [Map Property Reference](#) and [Functoid Reference](#), respectively.

Using BizTalk Editor, you can open a blank schema that contains no structure, you can open an existing XSD schema, or you can generate a schema from a non-XSD source. When you generate a schema from a non-XSD source, BizTalk Editor interprets the structure of the source and produces a schema that is an XSD representation of it. You can edit any records and fields that appear in the BizTalk Editor schema tree view, and then save the structure as a BizTalk schema.

## In This Section

- Planning for Schema Creation
- About Instance Messages
- About Schemas
- Using BizTalk Editor
- Creating Schemas
- Creating Schemas Using BizTalk Flat File Schema Wizard
- Testing Schemas
- Extending BizTalk Editor
- Considerations When Creating Schemas
- Known Issues with Schema Generation and Validation

## Planning for Schema Creation

You use schemas to validate instances of messages that are meant to conform to the schema, to define how instance messages in different formats (XML and non-XML) can be translated back and forth, and to define how XML instance messages with one structure can be transformed into XML instance messages with a different structure. For more information about the distinction between instance message translation and instance message transformation, see [Transformation vs. Translation](#).

The following table lists some of the questions that you need to answer in planning for schema creation in BizTalk® Editor.

Planning question	Recommendation
What schemas do I need to create?	Make a list of the business documents that you will be processing using Microsoft® BizTalk® Server 2006. Such a list might include, for example, a purchase order, an invoice, a shipping confirmation, and so on. The list might also include more than one of each business document, such as when the structure of a purchase order you receive from one trading partner is different than the structure of a purchase order you receive from another trading partner.
Are the documents	Add information to your list of documents regarding the format of each



send and receive already represented as XML?	business document you send and receive, be it XML or some other format such as a delimited or positional flat file format.
What starting points are available for the creation of the schemas on my list?	<p>Although sometimes necessary, creating schemas is more difficult than generating them from one of the supported sources. If your schema is already represented in XML Schema definition (XSD) language, no generation is necessary and you can simply open it in BizTalk Editor.</p> <p>If you have a well-formed XML instance message, a Document Type Definition (DTD) representation of your schema, or an XML-Data reduced (XDR) representation of your schema, you can automatically generate your schema. You may need to refine the generated schema using BizTalk Editor, but you will have saved yourself some work. For step-by-step instructions, see the procedure "To generate a schema from a non-XSD source" in Creating Schemas for XML Messages.</p> <p>If none of these starting points are available to you for one or more of the business documents on your list, you will need to create a new schema using BizTalk Editor and define its structure.</p>

## About Instance Messages

Microsoft® BizTalk® Server 2006 sends and receives instance messages, each of which typically represents one or more business documents such as a purchase order. An instance message is an instance of a message structure defined by one or more schemas. A schema, or a set of schemas being used together, defines what constitutes a valid instance message. For example, a purchase order might be defined to have several records within it, such as a ShipTo record, a BillTo record, an Items record, and so on. Each of these records can be defined to contain their own subrecords and fields. The corresponding schema defines the potential contents of these records and fields and the corresponding instance messages contain actual purchase orders that contain purchase order data structured according to the schema.

BizTalk Server can send and receive instance messages in an extensible variety of formats although one format, XML, has special significance as the format into which all message formats are translated for internal processing. A particular XML document uses a well-defined set of starting and ending tags, arranged hierarchically, to organize the data within the message and determine where one data item ends and another begins. One or more corresponding XML schemas define which tags are allowed within other tags, in what order, thereby governing the structure of conforming messages.

Another broad category of formats, known as flat file formats, are commonly used by legacy systems. These formats use some combination of delimiters (such as tabs) and fixed length fields to determine where one data item ends and another begins.

This section provides a high-level overview of the structure of these two types of messages that are commonly handled by BizTalk Server.

## In This Section

- Structure of an XML Message
- Structure of a Flat File Message

## Structure of an XML Message

In the context of Microsoft BizTalk Server 2006, an XML instance message is a valid hierarchy of XML tags that together constitute zero or more XML envelopes and one or more XML documents. For example, the following XML instance message consists of a single XML envelope (in regular font) that contains a single XML document (shown in bold type).

XML envelopes and the XML documents that they contain can be combined into valid XML instance messages in several different ways. The remainder of this section describes these different combinations.

## In This Section

- XML Message Envelopes
- Nested XML Message Envelopes

## XML Message Envelopes

XML envelopes serve two purposes within XML instance messages sent and received by Microsoft BizTalk Server 2006:

XML envelopes can contain data that supplements the data within the XML documents. This data can be promoted into the message context by the XML disassembler to provide easier access from a variety of BizTalk Server components. For outbound XML instance messages, the XML assembler can demote values from the message context into an envelope for inclusion in the instance message transmission.

XML envelopes can be used to combine multiple XML documents into a single, valid XML instance message. Without an envelope to wrap multiple documents within a single root tag, an XML instance message containing multiple documents would not qualify as well-formed XML.

A typical XML envelope (shown in bold type) contains both data and a tag used to delimit the one or more XML documents (shown in regular type) that it contains.

Less common, but still valid, an XML envelope (shown in bold type) need not contain any data or a tag for delimiting the XML documents (shown in regular type) that it contains.

In such cases, the XML envelope consists of nothing other than the starting and ending envelope tags.

## Nested XML Message Envelopes

XML envelopes can be nested to create complex document structures. Nested XML envelopes can occur in two forms, known as flexible and canonical. The following example shows the flexible form of enveloped documents, in which documents and envelopes (shown in bold type) can appear at the same level within an enclosing envelope.

The following example shows a similar instance message that conforms to the canonical form of enveloped documents, in which all documents appear at the same level within the innermost envelope.

Given an instance message in either of the forms described, the XML disassembler will produce document1, document2, document3, and document4. The message context of each of these documents includes the properties promoted from the corresponding document as well as the properties promoted within each of the enclosing envelopes. The following table shows the promoted properties that will be include in the message context of each unwrapped documents for both the flexible and canonical examples, given the property promotions specified in the first column for the various envelopes and documents.

<b>Specified property promotions</b>	<b>Resulting message context properties for flexible example</b>	<b>Resulting message context properties for canonical example</b>
envelope1: p1		
envelope2: p3	document1: p1, p2	document1: p1, p2, p3
document1: p2	document2: p1, p3, p4, p5	document2: p1, p3, p4, p5
document2: p4 and p5	document3: p1, p3	document3: p1, p3
document3: promotions no	document4: p1	document4: p1, p3
document4: promotions no		

## Structure of a Flat File Message

In the context of Microsoft BizTalk Server 2006, a flat file instance message is a text file that can contain three logical parts: a header, a body, and a trailer, in that order. Both the header and the trailer are optional. The following example shows a flat file instance message that consists of all three parts, with the body shown in bold type.

For the flat file disassembler to correctly distinguish the header, the body, and the trailer of a flat file instance message, you must create and configure a separate schema for each of them.

Within a particular part of a flat file instance message, different items of data are grouped into records, which themselves can contain subrecords and ultimately the individual items of data, known as fields. These records and fields are distinguished from each other using one of two

different basic methodologies. The first methodology, known as positional, defines each item of data to be of a pre-established length, with pad characters being used to bring a shorter item of data up to its expected length. The second methodology, known as delimited, uses one or more special characters to separate items of data from each other. This methodology avoids the need for otherwise superfluous pad characters, but introduces some special considerations when the data itself contains the character or sequence of characters being used as a delimiter.

The remainder of this section provides a high-level overview of how BizTalk Server handles headers, bodies, and trailers in flat file instance messages, and specifically, how it decides whether the optional parts are present, and how it separates the parts of inbound flat file instance messages and combines the parts of outbound flat file instance messages. This section also provides additional information about the differences between flat file instance messages that employ positional records and fields and flat file instance messages that employ delimited records and fields.

### In This Section

- Flat File Message Headers
- Flat File Message Bodies
- Flat File Message Trailers
- Flat File Messages with Positional Records
- Flat File Messages with Delimited Records
- Migrating Flat File Records

## Flat File Message Headers

The parsing of the optional flat file instance message header by the flat file disassembler is controlled by the flat file schema that you have configured in the **Header schema** design-time property of the flat file disassembler or the **XMLNORM.HeaderSpecName** message context property. If you have not specified a schema using one of these two methods, the flat file disassembler assumes that the flat file instance message does not contain a header.

For outbound flat file instance messages, you can configure the flat file assembler to produce a header by specifying the appropriate schema in its **Header Specification Name** design-time property or the **XMLNORM.HeaderSpecName** message context property. For more information about setting message context properties, see **Message Context Properties**.

Data found in inbound flat file instance message headers can be preserved and utilized in two different ways. First, flat file instance message headers can be saved in their entirety within the message context of the body for later restoration as the header of a corresponding outbound flat file instance message. You can use the **Preserve header** property of the receive pipeline to specify that the header should be preserved. And if a header is specified in the flat file assembler, the preserved header will be used on the outbound message.

Second, individual items of data from a flat file instance message header can be copied to the message context associated with the flat file message body by specifying property promotion for

one or more of the fields in the corresponding schema. For more information about promoting properties, see Promoting Properties.

## Flat File Message Bodies

A flat file instance message body, which is required, is what the flat file disassembler processes into one or more XML instance messages. To know what data to expect in an inbound flat file instance message body, you must configure the flat file disassembler with the flat file schema that corresponds to the body. You can specify the schema by using the **Document schema** design-time property of the flat file disassembler or the **XMLNORM.DocumentSpecName** message context property. Because flat file instance messages must have a body part, you must configure the appropriate schema using one of these two methods.

For outbound flat file instance messages, the flat file assembler can dynamically determine the appropriate flat file schema for the body of the instance message. The flat file assembler determines the appropriate schema from the message type, which is a combination of the target namespace and the name of the root element, both of which must be present in the XML version of the outbound message. Alternatively, you can explicitly configure the flat file schema to be used by configuring the **Document schema** design-time property of the flat file assembler or the **XMLNORM.DocumentSpecName** message context property.

Data found in inbound flat file instance message bodies can be copied to the corresponding message context by specifying property promotion in the flat file schema being used by the flat file disassembler to process the inbound instance message. Likewise, data in the message context can be copied back into outbound flat file instance messages by specifying property demotion in the flat file schema being used by the flat file assembler to process the outbound message.

## Flat File Message Trailers

As with flat file instance message headers, the parsing of the optional flat file instance message trailer by the flat file disassembler is controlled by the flat file schema that you have configured in the **Trailer schema** design-time property of the flat file disassembler or the **XMLNORM.TrailerSpecName** message context property. If you have not specified a schema using one of these two methods, the flat file disassembler will assume that the flat file instance message does not contain a trailer.

Unlike with flat file instance message headers, flat file instance message trailers can neither be saved and restored as a single unit, nor can they use property promotion to copy individual items of data to the message context associated with the flat file instance message body. However, a trailer can be added to an outbound flat file instance message by specifying the appropriate schema in the **Trailer schema** design-time property of the flat file assembler or the **XMLNORM.TrailerSpecName** message context property. The data that constitutes the variable portion of the trailer can be specified using property demotion from the message context of the flat file instance message body, or by specifying default or fixed values in the corresponding schema.

## Flat File Messages with Positional Records

Positional records within a flat file instance message contain individual fields (items of data) that are each of a predefined length. The fields are parsed according to these lengths. For example,

consider the following positional record from a flat file instance message, which contains a ship to address (the first line shows the number of characters reserved for each field).

A reasonable definition for this record in a flat file schema can be described as a positional record named shipTo that contains the following fields:

An attribute named country/region that is left-aligned, 10 characters in length, with a zero character offset.

An element named name that is left-aligned, 20 characters in length, with a zero character offset.

An element named street that is left-aligned, 20 characters in length, with a zero character offset.

An element named city that is left-aligned, 15 characters in length, with a zero character offset.

An element named state that is left-aligned, 2 characters in length, with a zero character offset.

An element named zip that is left-aligned, 5 characters in length, with a one character offset.

Given these record and field definitions, the flat file disassembler will produce the following XML equivalent of this record.

There are a number of considerations related to positional records that will affect how the record is parsed when received and constructed when sent, including:

The character used to fill the unused portion of each field, known as the pad character. For more information, see Field Padding.

An optional tag within the record, used to distinguish the record from other similar records. Tags usually occur at the beginning of the record but allowable anywhere within it. For more information, see Tag Handling in Positional Records. Positional records can be defined to have a tag or not have a tag, but once defined, the tag must be present or not, based on the definition.

How data is justified within a fixed length field, relative to the accompanying pad characters. For more information, see Field Justification.

Positional records nested within other positional or delimited records. For more information, see Nested Positional Records.

Positional records with field lengths specified as a specific number of bytes rather than a specific number of characters. For more information, see Position Counting in Bytes.

To help you better understand how to work with positional flat files, see the samples in the FlatFileReceive and FlatFileSend folders located at \Program Files\Microsoft BizTalk Server 2006\SDK\Samples\Pipelines\AssemblerDisassembler\.

## Flat File Messages with Delimited Records

Delimited records within a flat file instance message contain nested records and/or individual fields (items of data) that are separated by a predefined character or set of characters. The fields are parsed according to these separating delimiters. For example, consider the following delimited records from a flat file instance message, which contain two line items from a hypothetical purchase order:

A reasonable definition for this record in a flat file schema can be described as follows:

A delimited record named **items** with child delimiter (,), child order prefix, and the tag **ITEMS** (shown in bold type).

A delimited, repeating record named **item** with child delimiter |, child order infix, and the tag **ITEM** (shown in bold type).

An attribute named "partNum".

Given these record and field definitions, the flat file disassembler produces the following XML equivalent of these records.

There are a number of considerations related to delimited records that will affect how the record is parsed when received and constructed when sent, including:

The character or characters used to override the interpretation of delimiters so that they are treated as part of the data. For more information, see [Ways to Interpret Special Characters as Part of a Field Value](#).

An optional tag at the beginning of the record, used to distinguish the record from other similar records. For more information, see [Tag Handling in Delimited Records](#).

How data is justified within fields with minimum lengths, relative to the accompanying pad characters. For more information, see [Field Padding, Field Justification, and Minimum Field Lengths Within Delimited Records](#).

Positional records nested within other delimited records. For more information, see [Nested Positional and Delimited Records](#).

How data is justified within a fixed length field, relative to its accompanying pad characters. For more information, see [Field Justification](#).

Considerations concerning the positioning of delimiters relate to the data that they delimit. For more information, see [Child Order Considerations](#).

Preservation and suppression of delimiters when flat file messages are received and sent. For more information, see [Delimiter Preservation and Suppression](#).

To help you better understand how to work with delimited flat files, see the samples in the FlatFileReceive and FlatFileSend folders located at \Program Files\Microsoft BizTalk Server 2006\SDK\Samples\Pipelines\AssemblerDisassembler\.

## Migrating Flat File Records

Three schema annotations control delimiter handling in BizTalk 2000 and BizTalk 2002, two for parsing (**skip\_CR**, **skip\_LF**), and one for serializing (**append\_newline**). BizTalk 2006 interprets these annotations as follows when migrating records:

If the **skip\_CR** annotation has a value of **true** and the current delimiter is not carriage return (0x0D), BizTalk 2006 adds a carriage return to the current delimiter. For example, if the current delimiter were the pipe symbol (0x7C), the resulting delimiter is a pipe symbol followed by a carriage return (0x7C 0x0D). If the current delimiter is carriage return, it remains as a single carriage return regardless of the value of **skip\_CR**.

If the **skip\_LF** annotation has a value of **true**, BizTalk 2006 adds a linefeed character (0x0A) to the current delimiter. Notice that in the preceding case where the current delimiter is the pipe symbol (0x7C), a three character delimiter results (0x7C 0x0D 0x0A) if both **skip\_CR** and **skip\_LF** are **true**.

BizTalk 2006 ignores the setting of the **append\_newline** annotation.

While these interpretations of the annotations ensure the majority of cases migrate without difficulty, there are some cases where migration fails. For example, if **skip\_CR** and **skip\_LF** are both **true** and the current delimiter is the pipe symbol (0x7C), BizTalk 2000 and BizTalk 2002 accept all of the following as valid delimiters within a single set of records: 0x7C 0x0D 0x0A, 0x7C 0x0D, 0x7C 0x0A, and 0x7C. Records using such sets of delimiters cannot be migrated and require custom parser code in BizTalk 2006.

Although BizTalk 2006 has only one type of delimiter, it interprets the old annotations so that records migrate easily. If a BizTalk 2000 or BizTalk 2002 schema has values for **def\_record\_delim**, **def\_field\_delim**, **def\_subfield\_delim**, and these are referenced in **delimiter\_type** as **inherit\_record**, and so on, BizTalk 2006 retrieves the corresponding value and stores it locally.

In addition, BizTalk 2006 adds fields for tagged positional records without children.

## About Schemas

Microsoft® BizTalk® Server 2006 uses the XML Schema definition (XSD) language to define the structure of all messages that it processes, and refers to these definitions of message structure as schemas. With few exceptions, structured messages are the core of any application. These structured messages can take any form, large or small, and target a wide array of back-end systems and data stores. Systems that create and consume the structured messages frequently use different formats. Two of the most common formats for structured messages are XML and flat files.

BizTalk Editor is designed to simplify the process of defining a message schema and validating whether a particular message conforms to that schema. In the process of defining schemas and validating messages, you will likely perform some of the following tasks:

Create schemas for structured XML messages.



Create schemas for flat file messages.

Generate schemas from well-formed XML instance data.

Validate message conformance to a specific schema.

Perform design-time validation of schemas.

## In This Section

- Different Types of BizTalk Schemas
- Role of XSD
- XSD Resources on the Web
- BizTalk Representation of Schemas
- Schemas That Use Other Schemas
- Type Reuse and Derivations
- Schema Migration from Previous Versions of BizTalk Server
- Ways to Use Message Content to Control Message Processing
- Schema Validation in Visual Studio
- Instance Message Generation and Validation

## Different Types of BizTalk Schemas

BizTalk Server 2006 supports the following four types of schemas:

**XML schema.** An XML schema defines the structure of a class of XML instance messages. Because this type of schema uses XML Schema definition (XSD) language to define the structure of an XML instance message, and this is the intended purpose of XSD, such schemas use XSD in a straightforward way.

**Flat file schema.** A flat file schema defines the structure of a class of instance messages that use a flat file format, either delimited or positional or some combination thereof. Because the native semantic capabilities of XSD do not accommodate all of the requirements for defining the structure of flat file instance messages—such as the various types of delimiters that might be used for different records and fields within the flat file—BizTalk Server uses the annotation capabilities of XSD to store this extra information within an XSD schema. BizTalk Server defines a rich set of specific annotation tags that can be used to store all of the required additional information.

**Envelope schema.** An envelope schema is a special type of XML schema. Envelope schemas are used to define the structure of XML envelopes, which are used to wrap one or more XML

business documents into a single XML instance message. When you define an XML schema to be an envelope schema, a couple of additional property settings are required, depending on such factors as whether there are more than one root record defined in the envelope schema.

**Property schema.** A property schema is used with one of the two mechanisms that exist within BizTalk Server for what is known as property promotion. Property promotion is the process of copying specific values from deep within an instance message to the message context. From the message context, these values are more easily accessed by various BizTalk Server components. These components use the values to perform actions such as message routing. Promoted property values can also be copied in the other direction, from the more easily accessible message context back into the depths of the instance message, just before the instance message is sent to its destination. A property schema is a simple version of a BizTalk schema that plays a role in the process of copying promoted properties back and forth between the instance message and the message context.

The remainder of section provides additional information about these four types of schemas in BizTalk Server.

### In This Section

- XML Schemas
- Flat File Schemas
- Envelope Schemas
- Property Schemas

## Role of XSD

The XML Schema definition (XSD) language provides the underlying syntax of the message schemas defined within BizTalk. Although the tree views in BizTalk Editor and BizTalk Mapper use a BizTalk-specific graphical hierarchy of record and field nodes (among other types of nodes), each with its own set of properties, such hierarchies are constructed and persisted as XSD. BizTalk Editor provides a read-only XSD view in which you can see the XSD that is constructed as various nodes are added to or removed from the graphic representation of the schema in the tree view, and as the values of the properties associated with those nodes are changed. Although it is usually not necessary to understand the details of XSD to successfully construct simple schemas with BizTalk Editor, if you study the XSD changes as you make changes to the schema hierarchy in the tree view, you will learn to use XSD.

The schema annotation feature within XSD, with the extensive set of annotations defined by BizTalk Server, effectively allows XSD to be extended to support the necessary semantics for representing non-XML messages such as flat files.

**XSD Resources on the** Providing details of the XML Schema definition (XSD) language, or even adequately summarizing XSD, is beyond the scope of this document. For more detailed information, go directly to the XSD specifications and primer maintained on the World Wide Web Consortium (W3C) website. Specifically, the following tutorials are recommended, the first as a quick way to learn the basics, and the latter two as the definitive sources of information about XSD. For tutorials on XSD, see the following W3C pages:

XML Schema Part 0: Primer (<http://go.microsoft.com/fwlink/LinkId=16787>)

XML Schema Part 1: Structures (<http://go.microsoft.com/fwlink/LinkId=16788>)

XML Schema Part 2: Datatypes (<http://go.microsoft.com/fwlink/?LinkId=16789>)

It is recommended that you have a solid understanding of XSD before using BizTalk Editor to construct complex schemas.

## BizTalk Representation of Schemas

Although BizTalk schemas are ultimately represented and persisted in the XML Schema definition (XSD) language, they are represented as a graphical hierarchy of nodes when working in BizTalk Editor. The top of the hierarchy is always the **<Schema>** node, and the remaining types of nodes are used to build the schema so that it represents a particular message that is exchanged by using BizTalk.

BizTalk Editor provides a way to construct XSD schemas without needing to learn all of the intricacies of XSD syntax. When using the **Insert Schema Node** command on the **BizTalk** menu or the shortcut menu, the following choices for nodes to be inserted are available on the cascading menu.

Insert Schema Node menu choice	Description
<b>Child Record</b>	Inserts a <b>Record</b> node at the end of the sequence within the selected node. For more information about <b>Record</b> nodes, see Record Nodes.
<b>Child Field Attribute</b>	Inserts a <b>Field Attribute</b> node at the end of the selected <b>Record</b> or <b>Attribute Group</b> node. For more information about <b>Field Attribute</b> nodes, see Field Attribute Nodes.
<b>Child Field Element</b>	Inserts a <b>Field Element</b> node within the selected node. For more information about <b>Field Element</b> nodes, see Field Element Nodes.
<b>Sibling Record</b>	Inserts a <b>Record</b> node at the end of the sequence containing the selected node. For more information about <b>Record</b> nodes, see Record Nodes.
<b>Sibling Field Attribute</b>	Inserts a <b>Field Attribute</b> node at the end of the <b>Record</b> or <b>Attribute Group</b> node containing the selected node. For more information about <b>Field Attribute</b> nodes, see Field Attribute Nodes.
<b>Sibling Field Element</b>	Inserts a <b>Field Element</b> node at the end of the sequence containing the selected node. For more information about <b>Field Element</b> nodes, see Field Element Nodes.
<b>Sequence Group</b>	Inserts a <b>Sequence Group</b> node (<Sequence> in the tree view) at the end of the sequence within the selected node. For more information about <b>Sequence Group</b>

	nodes, see Sequence Group Nodes.
<b>Choice Group</b>	Inserts a <b>Choice Group</b> node (<Choice> in the tree view) at the end of the sequence within the selected node. For more information about <b>Choice Group</b> nodes, see Choice Group Nodes.
<b>All Group</b>	Inserts an <b>All Group</b> node (<All> in the tree view) as the only non-attribute child node of a <b>Record</b> node, replacing the default use of a <b>sequence</b> element within the <b>Record</b> node with the use of an <b>all</b> element. Before you can insert an <b>All Group</b> node, you must change the <b>Content Type</b> property of the containing <b>Record</b> node to <b>ComplexContent</b> . For more information about <b>All Group</b> nodes, see All Group Nodes.
<b>Attribute Group</b>	Inserts an <b>Attribute Group</b> node (<AttrGroup:attrGroupN> in the tree view, where <i>N</i> is a monotonically increasing numeral) at the end of the selected <b>Record</b> or <b>Attribute Group</b> node. For more information about <b>Attribute Group</b> nodes, see Attribute Group Nodes.
<b>Any Element</b>	Inserts an <b>Any Element</b> node (<Any> in the tree view) at the end of the sequence within the selected <b>Record</b> , <b>Sequence Group</b> , <b>Choice Group</b> , or <b>All Group</b> node. For more information about <b>Any Element</b> nodes, see Any Element Nodes.
<b>Any Attribute</b>	Inserts an <b>Any Attribute</b> node (<AnyAttribute> in the tree view) at the end of the sequence within the selected <b>Record</b> or <b>Attribute Group</b> node. For more information about <b>Any Attribute</b> nodes, see Any Attribute Nodes.

In many cases, adding a single node in BizTalk Editor results in the addition of multiple nested elements within the corresponding XSD representation of the schema. Because these nested elements can have complex syntax, using BizTalk Editor to graphically arrange nodes is a much less error-prone approach to constructing XSD schemas than hand-editing the XSD. Another factor to consider is that always using BizTalk Editor to construct XSD schemas results in a more controlled subset of XSD being used in the schema descriptions.

Overall, BizTalk Editor combines a simplified approach to constructing XSD schemas using the generic concepts of Records and Fields with a more explicit control of particular XSD constructs, such as **sequence**, **choice**, **any**, and **anyattribute** elements.

Each type of node has a unique set of properties that can be configured in the Visual Studio Properties window. In general, these properties correspond to attributes on the XSD elements in the corresponding XSD representation of the schema. For more information about node properties, see Node Properties and **Schema Property Reference**.

This section describes the types of nodes used within BizTalk Editor, briefly discusses their properties, and provides links to reference information about their properties.

## In This Section

- Nodes That Correspond Directly to Message Instance Data and Structure

- Nodes That Control Instance Message Structure and Variations
- Nodes That Simplify Schema Creation
- Nodes That Provide Wildcard Capabilities
- Node Properties

## Nodes That Correspond Directly to Message Instance Data and Structure

Some of the node types that you use to create schemas in BizTalk Editor correspond directly to elements and attributes in XML representation of instance messages governed by the schema (for other instance message formats, such as flat file formats, this correspondence only exists after translation from the other format and before translation to the other format). These node types are **Record** nodes (including root **Record** nodes), **Field Element** nodes, and **Field Attribute** nodes.

The values you give to the **Node Name** property of **Record** and **Field Element** nodes specify the name of the corresponding element in XML instance messages governed by the schema. Likewise, the values you give to the **Node Name** property of **Field Attribute** nodes specify the name of the corresponding attribute in XML instance messages governed by the schema.

The remainder of this section provides more information about this class of nodes, with root **Record** nodes receiving separate treatment due to their special role in schemas.

### In This Section

- Root Nodes
- Record Nodes
- Field Element Nodes
- Field Attribute Nodes

## Root Nodes

In BizTalk Editor, child nodes of the **Schema** node are known as **Root** nodes. **Root** nodes are a special type of **Record** node, and have a few more properties than regular **Record** nodes. The **Root** node represents the type of document described by the schema, and can be renamed as appropriate. For example, you can rename the **Root** node so that it describes the type of message that the schema represents, such as purchaseOrder, orderAcknowledgment, or shipNotice.

When you create a new XML schema in BizTalk Editor, the **Schema** node and one **Root** node are created automatically. You can create additional **Root** nodes as children of the **Schema** node; this enables you to create a library of schemas within a single XML Schema definition (XSD) language representation. For example, you can create a library of schemas to describe the various schemas of messages related to sending purchase orders, naming the various root nodes purchaseOrder, orderAcknowledgment, and shipNotice.

## XSD representation

The following example shows, in bold type, the lines in the XSD representation of the schema that correspond to the **Root** node in the tree view of the schema.

**Root** nodes in BizTalk Editor represent the main element in a corresponding XML instance of the message in question. For example, if the **Root** node of a particular schema is renamed to `purchaseOrder`, the corresponding XSD representation has the following high-level structure.

A corresponding XML instance message must have the following basic structure.

## Record Nodes

In BizTalk Editor, you use a **Record** node to represent a collection of information, the individual items of which can be:

- Simple types of information, such as strings and numbers, represented as child field nodes. These child field nodes can be either **Field Element** nodes or **Field Attribute** nodes. For additional information about these two types of field nodes, see [Field Element Nodes](#) and [Field Attribute Nodes](#).

- Complex types of information, represented as child **Record** nodes or as a group node (**Sequence Group** node, **Choice Group** node, or **All Group** node).

- Any unexamined type of information, represented as child **Any Element** or **Any Attribute** nodes.

- Groups of attributes represented by an **Attribute Group** node.

When you insert a new child node into a **Record** node, the child node is always inserted at the end of the current child nodes. Within the XML Schema definition (XSD) language representation, new elements are added to the end of their corresponding areas, meaning that nonattribute elements are added to the end of the elements within the **sequence**, **choice**, **all**, or **group** element, and attribute elements are added to the end of any other attribute elements, all of which occur after the **sequence**, **choice**, **all**, or **group** element.

## XSD representation

When first inserted, the XSD representation of a new **Record** node consists of only three lines, as shown in the following example.

When any child node other than one of the three attribute nodes (**Field Attribute**, **Attribute Group**, and **Any Attribute**) is added to a **Record** node, by default it is placed within a **sequence** element within the **complexType** element. The **sequence** element is added when the first nonattribute child node is added, and removed if all the nonattribute child nodes are deleted. All three types of attribute nodes are added within the **complexType** element, but outside and after any **sequence** element.

The **sequence** element within which nonattribute child nodes are added can also be a **choice** or **all** element if you change the **Group Order Type** property of the corresponding node in the schema tree to **Choice** or **All**, respectively.

In the following example, the **Record** node has been renamed shipTo. The locations within the **Record** node where attribute and nonattribute nodes are added are shown in brackets.

## Field Element Nodes

In BizTalk Editor, you use **Field Element** nodes to describe items of information that are simple in nature, such as strings and numbers. Further, they are used when the information in question appears as the content of an XML element in an actual instance of a message, as opposed to appearing as the value of an attribute associated with an XML element. For additional information about information that is stored as attribute values, see Field Attribute Nodes.

For any given item of information in an XML message, where item of information means a single discrete simple type, such as a string or number, there is always a question regarding whether that information should be represented as the attribute of an element, or as a subelement of that element. As a general rule, representing an item of information as an attribute tends to be more appropriate when the possible values are discrete, few in number, and tend to modify the semantics of the element itself. Representing an item of information as a subelement tends to be more appropriate when the possible values can repeat a variable number of times, are likely to have more widely ranging values, might be long, as in long strings, and are one of several sibling values where their order is relevant. If you are just creating a schema for an existing type of XML document, your choice of using a **Field Element** node or a **Field Attribute** node for a particular item of information has already been made for you, and you must use the node that matches the XML.

### XSD representation

When a **Field Element** node is inserted into a **Record** node, it is inserted at the end of any other child nodes within the **sequence** element in the **Record** node. The following example shows a new **Field Element** node, in bold, inserted at the end of the **sequence** element in a **Record** node (with nodes named to clarify their identity).

## Field Attribute Nodes

In BizTalk Editor, you use **Field Attribute** nodes to describe items of information that are simple in nature, such as strings and numbers. Further, they are used when the information in question appears as the value of an attribute in an actual instance of a message, as opposed to appearing as the content of an XML element. For additional information about information that is stored as element content, see Field Element Nodes.

Although the most straightforward use of **Field Attribute** nodes is as child nodes of **Record** nodes, they can also be used as child nodes of **Attribute Group** nodes. In the latter case, the **Field Attribute** nodes that are children of an **Attribute Group** node are available as attributes of any **Record** node that includes that **Attribute Group** node. For more information about **Attribute Group** nodes, see Attribute Group Nodes.

For any given item of information in an XML message, where item of information means a single discrete simple type, such as a string or number, there is always a question regarding whether that information should be represented as the attribute of an element, or as a subelement of that element. As a general rule, representing an item of information as an attribute tends to be more appropriate when the possible values are discrete, few in number, and tend to modify the

semantics of the element itself. Representing an item of information as a subelement tends to be more appropriate when the possible values can repeat a variable number of times, are likely to have more widely ranging values, might be long, as in long strings, and are one of several sibling values where their order is relevant. If you are creating a schema for an existing type of XML document, your choice of using a **Field Element** node or a **Field Attribute** node for a particular item of information has already been made for you, and you must use the node that matches the XML.

## XSD representation

When a **Field Attribute** node is inserted into a **Record** node, it is inserted at the end of any other child nodes in the **Record** node. This includes being inserted after the **sequence**, **choice**, or **all** element containing any nonattribute nodes, and after any attribute nodes that were previously inserted. The following example shows a new **Field Attribute** node, in bold, inserted at the end of a **Record** node (with nodes named to clarify their identity).

## Nodes That Control Instance Message Structure and Variations

Some of the node types that you use to create schemas in BizTalk Editor control the structure of, and variations within, instance messages. You use **Sequence Group** nodes to specify that a sequence of elements must occur in a specific order in the corresponding location in an instance message. You use **Choice Group** nodes to specify that one element from a collection of elements can occur in the corresponding location in an instance message. You use **All Group** nodes to specify that all of the specified elements can occur in any order, but only once, at the corresponding location in an instance message. **<Equivalent>** nodes and their child nodes are displayed in the schema tree to indicate locations in instance messages where derivation-based polymorphism is in effect, allowing one of many related complex data types to occur in the corresponding location in an instance message.

The remainder of this section provides additional information about this class of nodes.

### In This Section

- Sequence Group Nodes
- Choice Group Nodes
- All Group Nodes
- <Equivalent> Nodes and Their Child Nodes

## Sequence Group Nodes

In BizTalk Editor, you can insert a **Sequence Group** node to contain other nodes that must appear in an instance message in the same order in which they appear within the **Sequence Group** node. The contained nodes must be nodes that correspond to XML elements, but cannot be nodes that correspond to XML attributes.



You may want to add a **Sequence Group** to declare a global element group.

You may need to create a schema for XML as follows.

Because GroupItem1 and GroupItem2 exist in both cases, you may declare a global sequence group that is a child of both Record1 and Record2. For step-by-step instructions about how to declare a global sequence group, see [Creating References to Another Node or Type](#).

A user can change the hidden group to be a **Choice Group** node or an **All Group** node (so it is not necessarily a **Sequence Group** node) by changing the **Group Order Type** property.

### XSD representation

When a **Sequence Group** node is inserted into a **Record** node, it is inserted at the end of any other child nodes within the **sequence**, **choice**, or **all** element in the **Record** node. The following example shows a new **Sequence Group** node, in bold type, inserted at the end of the **sequence** element in a **Record** node (with nodes named to clarify their identity).

## Choice Group Nodes

In BizTalk Editor, you can insert a **Choice Group** node to contain other nodes (or entire subtrees of nodes), only one of which can appear in an instance message. A given instance message, if valid, will have only one of the choices present. The contained nodes must be nodes that correspond to XML elements, but cannot be nodes that correspond to XML attributes.

### XSD representation

When a **Choice Group** node is inserted into a **Record** node, it is inserted at the end of any other child nodes within the **sequence**, **choice**, or **all** element in the **Record** node. The following example shows, in bold type, how a new **Choice Group** node is represented in the XML Schema definition (XSD) language as a **choice** element inserted at the end of the **sequence** element in a **Record** node (with nodes named to clarify their identity).

By default, the **choice** element is given a **minOccurs** attribute value of zero (0), indicating that none of the choices need occur. You can change this value in the Visual Studio Properties window when the **Choice Group** node is selected in the schema tree view.

The following example shows the same **choice** element with the XSD **element** elements corresponding to two subordinate **Record** nodes.

In this example, two sibling **Record** nodes are used to describe the fact that an instance message will either have a record with United States address information in it, or a record with worldwide address information in it. Further, the **minOccurs** and **maxOccurs** properties of the **Choice Group** node have both been set to one (1) in the Visual Studio Properties window, resulting in the *minOccurs* and *maxOccurs* attributes of the **choice** element being set to one (1) in the XSD representation.

## All Group Nodes

In BizTalk Editor, you can insert an **All Group** node to contain other nodes that will appear zero or one time, in any order. In XML Schema definition (XSD) language, the **All group** has more usage limitations than **Sequence** and **Choice** groups, which translates to few situations within BizTalk Editor where you will be able to create an **All Group** node.

To use an **All Group** node in BizTalk Editor, you need to follow some extra steps: The easiest way to create an **All Group** node is to change the value of the **Group Order Type** property of the parent **Record** node to **All**. This ensures that all of the subordinate nodes of the **Record** node are contained within the **All Group** node.

Another way to use an **All Group** node in BizTalk Editor begins with inserting a new **Record** node. After inserting the new **Record** node, change its **Content Type** property to **ComplexContent**. Then you can insert an **All Group** node as a child of the **Record** node. This is required because the **All Group** can only be inserted when inheritance is involved. By specifying that the containing **Record** node contains complex content, its data type becomes based on the data type **xs:anyType**, derived by extension.

### XSD representation

An **All Group** node can be inserted into a **Record** node, but only if it is the only non-attribute child node of that **Record** node. The following example shows, in steps, how a new **All Group** node is represented in the XML Schema definition (XSD) language as an **all** element as the steps in BizTalk Editor are performed (with nodes named to clarify their identity).

## <Equivalent> Nodes and Their Child Nodes

The **<Equivalent>** node and its children are used in the schema tree to display occurrences of complex data type polymorphism. When you derive one complex data type from another complex data type, polymorphism within XSD allows either of these data types to occur in instance messages in locations for which the base complex data type has been specified. During schema validation, the fact that one of multiple possible complex data types is allowed at a particular location is represented implicitly by the base complex data type name associated with the **base** attribute of the **extension** or **restriction** elements of the derived complex data types. To make this polymorphism more obvious in the schema tree, the **<Equivalent>** node and its child nodes are displayed explicitly.

The **<Equivalent>** node is displayed as a child node of occurrences of the base complex data type, indicating that there are multiple complex data types that could occur at that position in an instance message. The child nodes of the **<Equivalent>** node are displayed as the value of the **name** attribute of the corresponding **complexType** element in the XSD representation of the polymorphism, displayed within angle brackets (<name>).

The **<Equivalent>** node and its children each have only two properties associated with them, as follows:

**<Equivalent>** node: **Node Name** and **Base Type**

Child nodes: **Node Name** and **Derivation Type**

## XSD representation

There is no direct XSD representation of the **<Equivalent>** node and its children. This node is used within the schema tree to make complex data type polymorphism more visible and obvious.

## Nodes That Simplify Schema Creation

The nodes that simplify schema creation is a miscellaneous category for two node types that defy categorization into the other categories, and which simplify the creation of schema. The first such node, the **Schema** node, corresponds to the **schema** element in XML Schema definition (XSD) language, which is the element that serves as the outermost element in all XSD schemas.

The other simplification node plays a more significant role in this regard. The **Attribute Group** node corresponds directly to the **attributeGroup** element in XSD, and provides a mechanism for defining a group of related attributes, which always occur together, in a single location and then using that group definition in multiple locations throughout the schema. The remainder of this section provides additional information about these nodes.

### In This Section

- Schema Node
- Attribute Group Nodes

## Schema Node

In BizTalk Editor, the top of the schema hierarchy is always the **Schema** node, which cannot be renamed. The **Schema** node corresponds to the **schema** element in the XML Schema definition (XSD) language representation of the schema.

In general, the properties of the **Schema** node correspond to the attributes of the **schema** element in the XSD representation of the schema. For general information about node properties, see Node Properties. For reference information about the properties of the **Schema** node, see **Schema Node Properties**.

When you create a new XML schema in BizTalk Editor, the **Schema** node and one **Root** node are created automatically.

## XSD representation

The following example shows, in bold type, the lines in the XSD representation of the schema that correspond to the **<Schema>** node in the tree view of the schema.

## Attribute Group Nodes

In BizTalk Editor, you can add an **Attribute Group** node to a **Record** node or to another **Attribute Group** node to contain a group of attributes that you expect to use in more than one **Record** node. Adding an **Attribute Group** node to another **Attribute Group** node achieves attribute group nesting. This allows you to define a group of attributes in one place that can be

used in multiple **Record** or **Attribute Group** nodes. Subsequent modifications to the attribute group will propagate to all of the nodes with which that attribute group is associated. This is true regardless of the node context in which the modifications are made.

When initially creating an **Attribute Group** node, you simply insert it into one of the **Record** or **Attribute Group** nodes in which it will be used, and optionally change its name in its **Group Reference** property. There are two ways to use the same attribute group in another **Record** or **Attribute Group** node:

You can copy the existing **Attribute Group** node and then paste it into that other **Record** node.

You can insert a new **Attribute Group** node into that other **Record** node, and then set the **Group Reference** property of the new **Attribute Group** node to reference an existing **Attribute Group** node.

Thereafter, you can modify the **Attribute Group** node—for example, by adding or deleting a **Field Attribute** node—in the context of any **Record** or **Attribute Group** node into which you pasted it. That change will propagate to all other **Record** or **Attribute Group** nodes with which the attribute group is associated.

It would be pointless to add an **Attribute Group** node without adding at least one relevant node to it, where relevant nodes include **Field Attribute** nodes, **Any Attribute** nodes, and (nested) **Attribute Group** nodes. In fact, an attribute group that contains only a single attribute is somewhat ill-conceived, unless you are making a point of planning for the addition of more attributes in the future.

**Attribute Group** nodes can be nested, allowing more possibilities in how groups of attributes can be constructed and combined. **Attribute Group** nodes can also contain the **Any Attribute** node, allowing an attribute group to contain wildcard character capabilities with respect to the attribute instances it can accommodate.

## XSD representation

When an **Attribute Group** node is first added to a **Record** node or to another **Attribute Group** node, two distinct areas of the corresponding XML Schema definition (XSD) language representation of the schema are affected. In the following example, a new **Attribute Group** node, in bold, has been added to an existing **Record** node that already contains an existing **Field Element** node.

Note how the **attributeGroup** element within the XSD representation of the **Record** node references a global **attributeGroup** element that is added as a child of the **schema** element. This global definition of the attribute group within the XSD representation of the schema allows the attribute group to be referenced in multiple locations throughout the schema.

## Nodes That Provide Wildcard Capabilities

BizTalk Editor provides a direct mapping to the XSD concepts that provide wildcard character capabilities for elements and attributes. The **Any Element** node maps directly to the XML Schema definition (XSD) language **any** element and the **Any Attribute** node maps directly to the XSD

**anyAttribute** element. The remainder of this section provides additional information about these nodes.

### In This Section

- Any Element Nodes
- Any Attribute Nodes

## Any Element Nodes

In BizTalk Editor, you can use an **Any Element** node to indicate a location within an instance message where unknown elements may appear. This accommodates situations in which you know that some element might appear at a particular location within an instance message, but you do not know the name of the element, or how complicated it might be. If you place an **Any Element** node at the appropriate location within the schema, BizTalk can process such unknown portions of a message. The only requirement is that the corresponding XML is well-formed.

### XSD representation

When an **Any Element** node is added to a **Record** node, or to another node to which it can be added such as a **Sequence Group**, **Choice Group**, or **All Group** node, a single XML tag is added to the corresponding XML Schema definition (XSD) language representation of the schema. In the following example, a new **Any Element** node, whose XSD representation is shown in bold type, has been added to an existing **Record** node that already contains a **Field Element** node.

Assuming that the **Process Contents** property of the **Any Element** node is set to **Skip**, within an instance message governed by this schema fragment, an **ExistingRecord** element is expected to contain an **ExistingFieldElement** element containing string data, followed by any single element of arbitrary complexity.

## Any Attribute Nodes

In BizTalk Editor, you can use an **Any Attribute** node to indicate a (known) element within an instance message for which zero or more unknown attributes may appear. This accommodates situations in which you know that a particular element will be present at a particular location within an instance message, but you are not sure exactly what attributes that element might include. If you place an **Any Attribute** node within the **Record** node associated with the relevant element, BizTalk can process that element, with the only requirement being that any associated attributes are syntactically correct (attributeName="attributeValue").

### XSD representation

When an **Any Attribute** node is added to a **Record** node or to an **Attribute Group** node, a single XML tag is added to the corresponding XML Schema definition (XSD) language representation of the schema. In the following example, a new **Any Attribute** node, whose XSD representation is shown in bold, has been added to an existing **Record** node that already contains a **Field Element** node.

In the preceding example, the XSD representation of the new **Any Attribute** node adds an **anyAttribute** element to the end of the containing (**Record** node) **element** element, outside the **sequence** element and within the **complexType** element. This is where all **attribute** elements, other than those with an **Attribute Group** node, are added to their containing **element** elements.

Now, and assuming that the **Process Contents** property of the **Any Attribute** node is set to **Skip**, within an instance message governed by this schema fragment, an **ExistingRecord** element is expected, and it can contain any attributes so long as they are well-formed with respect to XML syntax. (To conform to the XSD fragment in this example, it must also contain the **ExistingFieldElement** element as well.)

## Node Properties

In BizTalk Editor, you examine and set node properties in the Visual Studio Properties window. As you select different types of nodes in the schema tree view, different sets of properties are displayed in the Properties window. As is standard in Visual Studio .NET, these properties can be displayed either in categories or alphabetically with no indication of their categories. Use the standard buttons near the top of the Properties window to toggle this setting.

Node properties, especially when set to values other than their defaults, are generally represented in the XML Schema definition (XSD) language as attributes and attribute values associated with the corresponding element. For example, when properties are set for the **Min Occurs** and **Max Occurs** properties, which are available for several different node types, the values that are set are used as the values of the **minOccurs** and **maxOccurs** attributes, respectively, associated with the element that represents the node for which the **Min Occurs** and **Max Occurs** properties are being set.

The BizTalk Server 2006 Programmer's Reference, in this documentation set, contains a reference section for the properties of the various node types, organized by category and alphabetically. The following topics summarize the properties associated with each node type:

### Schema Node Properties

#### Record Node Properties

#### Field Element Node Properties

#### Field Attribute Node Properties

#### Sequence Group Node Properties

#### Choice Group Node Properties

#### All Group Node Properties

#### Attribute Group Node Properties

#### Any Element Node Properties

#### Any Attribute Node Properties

## Equivalent Node Properties

### Equivalent Child Node Properties

The topic **Node Properties — Alphabetical Listings** contains all of the individual reference topics for each node property, some of which apply to various types of nodes. The individual reference topics are categorized according to whether they are basic properties that apply to all types of schemas, or a specialized properties that are associated with a schema editor extension, such as the flat file extension. Within these categories, they are listed alphabetically.

BizTalk Editor uses the Visual Studio .NET Properties window to enable you to examine and set the properties of the nodes in the schema tree. This section describes some characteristics of working with properties in the Properties window, including special considerations for the **Node Name** property, an explanation of the interdependencies between properties, and information about the maximum lengths allowed for certain properties or types of properties.

The remainder of this section provides additional information about particular, special node properties and other information that applies generally to node properties.

### In This Section

- Node Name Property
- Property Interdependencies
- Additional Flat File Properties

## Node Name Property

As you use BizTalk Editor to insert nodes into the schema tree, some nodes are meant to be renamed and others are not. Essentially, you can and should rename **Record** nodes, **Field Element** nodes, and **Field Attribute** nodes. The names that you give to these nodes will become the names of the XML elements and attributes in the message that the schema defines.

In the schema tree, the nodes that you cannot rename are shown in the form of XML tags; that is, with the less than (<) and greater than (>) signs. For example, the **Schema** node, **Choice Group** nodes, **Any Element** nodes, and **Any Attribute** nodes are represented in the schema tree with the names <Schema>, <Choice>, <Any>, and <AnyAttribute>, respectively. The **Node Name** property for such nodes is read-only.

Within a given **Record** node, you cannot have two **Field Attribute** nodes with the same name. However, you can have more than one **Field Element** node or **Record** node with the same name as child nodes of the same **Record** node, as long as they all have the same data type (as specified by their **Data Type** property for **Field Element** nodes or their **Data Structure Type** for **Record** nodes).

When you give names to **Record** nodes, **Field Element** nodes, and **Field Attribute** nodes, use names that are descriptive of the role of that element or attribute within the message being defined by the schema. For example, `FirstName` is probably a good choice for the name of a **Field**

**Element** node that will be used to store the first name of someone in an address structure. In an XML instance message where the first name James occurs, the corresponding element would look like the following.

When you are renaming **Record** nodes, **Field Element** nodes, and **Field Attribute** nodes, you should be aware that not all characters are allowed in node names. For information about these disallowed characters, see Which Node Name Characters Get Encoded. Although BizTalk Editor allows you to use disallowed characters by encoding them, it is often simpler to avoid such characters altogether. For information about how disallowed characters are encoded, see How Node Name Characters Get Encoded.

In addition to the characters that are disallowed in node names, unless they are encoded in the XSD representation of the schema, you should not use C# reserved words as the names of any root nodes in the schema tree (unless you provide a valid **RootNode TypeName** property value) or as schema file names.

### In This Section

- Which Node Name Characters Get Encoded
- How Node Name Characters Get Encoded

## Property Interdependencies

As you use BizTalk Editor, and specifically the Visual Studio .NET Properties window, to change the values of properties, you will notice that there are extensive interdependencies between properties. Sometimes a particular setting for one property will cause other properties to be automatically cleared, become enabled or disabled, or even appear or disappear entirely from the Properties window. These interdependencies are too numerous to cover. However, the following list provides some common examples to give you an idea of how they work:

When setting the properties of a **Field Element** node or **Field Attribute** node for which a data type is being derived from a simple type by using the restriction mechanism, an entire new category of properties becomes available: **Restriction**. Further, the properties in this new category are enabled or disabled based on whether the base data type is of a string type or a numeric type. For more information about this form of simple type derivation, see Simple Type Derivation Using the Restriction Mechanism.

When setting the properties of a **Field Element** node or **Field Attribute** node for which a data type is being derived from a simple type by using either the list or union mechanism, the **Base Data Type** property is changed to either the **Item Type** property or the **Member Types** property, respectively. In the latter case, the corresponding drop-down list is modified to include check boxes, allowing multiple types to be selected. For more information about these forms of simple type derivation, see Simple Type Derivation Using the List Mechanism and Simple Type Derivation Using the Union Mechanism.

To expose the properties associated with flat file schemas, you must set the **Schema Editor Extensions** property of the **Schema** node to include the **Flat File Extension**. The custom properties associated with other editor extensions, such as the EDI extension, are exposed in the same way: by choosing the corresponding extension using the **Schema Editor Extensions** property.



This list includes examples that are meant to illustrate the types of property interdependencies that you will see when working within the Properties window, but it is not meant to be an exhaustive list of such interdependencies.

## Additional Flat File Properties

The following table lists additional flat file node properties that do not appear in the Schema Editor. Using these properties requires hand editing the schema file in a text editor.

Property	Values	Default Value	Description
suppress_empty_nodes	<b>true</b> or <b>false</b>	<b>false</b>	Indicates whether or not to remove empty XML nodes after the parser generates XML instance data.
generate_empty_nodes	<b>true</b> or <b>false</b>	<b>true</b>	Generate empty nodes for records that exist in the XML instance data.
parser_optimization	<b>speed</b> or <b>complexity</b>	<b>speed</b>	Optimizing for speed decreases the parsing time but at the cost of dealing with some ambiguities in data. Optimizing for complexity handles a wider range of ambiguities but at the cost of processing speed.
lookahead_depth	Any positive integer; zero (0) indicates infinite lookahead.	3	How far to look ahead for matching data.
allow_early_termination	<b>true</b> or <b>false</b>	<b>false</b>	Indicates whether positional records can terminate early ( <b>true</b> ) or must contain data for all record fields ( <b>false</b> ).
early_terminate_optional_fields	<b>true</b> or <b>false</b>	<b>false</b>	<p>Enable early termination of optional trailing fields (<b>true</b>). If the existing schema without this annotation is opened in the BizTalk Editor, this annotation will be added to it with the default value set to (<b>false</b>).</p> <p><b>Note</b> The early_terminate_optional_fields annotation only takes effect if the allow_early_termination is set to (<b>true</b>).</p>

All of these properties are attributes of the **/annotation/appinfo/schemaInfo** element.

When **parser\_optimization** is set to **complexity**, you may have validation failures against a schema when there are many optional nodes in the same group or record. You may need to set **lookahead\_depth** to zero (0) to avoid validation errors.

## Schemas That Use Other Schemas

When your schemas become large and complex, or when the schemas that represent your different types of instance messages have some portions in common, it can be useful to combine smaller schemas into the schemas that ultimately define the structure of the instance messages you plan to exchange with trading partners. For example, you might have multiple message types that require a shipping address to be expressed within them. You can define the structure of a shipping address in a single schema, and then use that schema within other schemas that define, for example, Order, Invoice, and Shipping Notice message schemas.

XML Schema definition (XSD) language provides three related mechanisms for using multiple schemas together that BizTalk Editor supports. The following table summarizes the characteristics of these mechanisms, as defined by XSD.

Multischema mechanism	Usage scenario
Import	<p>Accesses and uses types defined in the imported schema.</p> <p>Must use types in the imported schema as is, or derive new types from them; no type modification allowed.</p> <p>Provides a mechanism for using types defined in other namespaces. Indeed, an imported schema must have a target namespace that is different from the importing schema.</p> <p>Uses the <b>import</b> element and its <b>namespace</b> and <b>schemaLocation</b> attributes to reference the other schema.</p>
Include	<p>Accesses and uses types defined in the included schema.</p> <p>Must use types in the included schema as is, or derive new types from them; no type modification allowed.</p> <p>The included schema must be in the same target namespace as the including schema, or the target namespace of the included schema must be empty.</p> <p>Uses the <b>include</b> element and its <b>schemaLocation</b> attribute to reference the other schema.</p>
Redefine	<p>Accesses and uses types defined in the redefined schema.</p> <p>Can use types in the redefined schema as is, derive new types from them, or specify modifications to them.</p> <p>The redefined schema must be in the same target namespace as the redefining</p>

	<p>schema, or the target namespace of the redefined schema must be empty.</p> <p>Uses the <b>redefine</b> element and its <b>schemaLocation</b> attribute to reference the other schema. Any type redefinitions are specified with the <b>redefine</b> element.</p>
--	---

To use a type defined in one schema (Schema1) within another schema (Schema2), you must provide a reference to Schema1 within Schema2. To do so, use the **Imports** property of the **Schema** node in Schema2. When you click the ellipsis (...) button in the **Imports** property field, the **Imports** dialog box opens. In the **Import New Schema as** drop-down list, select **XSD Import**, **XSD Include**, or **XSD Redefine**. Then click **Add** to open the **BizTalk Type Picker** dialog box and browse within your BizTalk project to select **Schema1**.

For detailed instructions about these steps, see [Creating Schemas That Use Other Schemas](#).

When you use the **Imports** dialog box to import, include, or redefine another schema, one or more of the XSD elements **import**, **include**, and **redefine** is added to the XSD representation of your schema, including the appropriate attributes and attribute values. Further, in the case of the **import** element, a prefix declaration for the namespace of the other schema is added to the **schema** element.

All global types (such as **ComplexTypes**, **SimpleTypes**, element groups, attribute groups) in an imported/included/redefined schema are automatically available for use within the schema in which the former schema is imported, included, or redefined. For example, global **ComplexTypes** defined in an imported/included/redefined schema are added to the drop-down list of the **Data Structure Type** property for all of the **Record** nodes in the importing, including, or redefining schema.

## Type Reuse and Derivations

Within XML Schema definition (XSD) language, complex global types provide a mechanism for defining a structured data type that can be reused, and potentially redefined, at various locations within your schema. Perhaps the most classic example is an address structure that includes a name, street, city, state, and so on. Further, the name itself might be a structure that includes first, middle, and last name strings. If this complex structure is defined globally, you can use it in multiple locations within your schema, such as for both a shipping address and a billing address.

XSD also provides mechanisms for deriving one type from another. This includes both simple content types and complex content types. For example, a new type can be derived from a simple string type (such as, `xs:string`) such that the new type allows only a few particular strings as legal values. This type of derivation is known within XSD as derivation by restriction because the values allowed by the derived type are more restrictive than the values allowed by the base type.

An example of a derivation involving a complex type can be seen in the address type previously suggested. Suppose that the address type is designed to accommodate the addresses within a particular country/region, where the country/region itself is assumed in the address. To extend such an address type to handle international addresses, you can derive a new type from the original address type and then include additional information in the derived type, such as the

country/region. This type of derivation is known within XSD as derivation by extension because the derived type has extended the base type.

This section describes type reuse and the ways in which you can use derivation to redefine types as they are reused.

**In This Section**

- Complex Global Type Definition and Naming
- Ways to Use Complex Global Types
- Simple Type Derivation