



**THE
ENABLER
GROUP**

BizTalk Utilities Adapter for Messaging

The Adapter is aimed at providing BizTalk Server the ability to talk to the following Messaging Applications:

- Java Messaging Services (JMS)
- IBM MQSeries

Redesigned and developed to allow for integration into the new BizTalk Server Adapter Framework.

The Adapter incorporates new interface to the Java Messaging Service API, and is no longer COM based as with the previous version of the Messaging Adapter.

Developed in C#.NET it provides a fast optimized mechanism for communicating with Messaging Application from BizTalk Server.

The Adapters are fully integrated into the Visual Studio .NET environment providing Developers with an enhanced Visual Studio .NET Experience when configuring.

The Adapter is extremely easy to install and configure. All tasks performed when configuring the Adapter is Wizard Driven.

System Requirements

The following are the minimum software requirements for the Adapter:

- Microsoft Enterprise Instrumentation Framework
- Microsoft BizTalk Server

Enterprise Instrumentation Framework (EIF)

The Adapter utilizes EIF for writing event and tracing information to the Application Event Log and Windows Trace Files.

The Microsoft Enterprise Instrumentation framework provides unified management, eventing, and diagnostic tracing services for enterprise applications in a production environment. Enterprise Instrumentation enables developers to consistently instrument enterprise applications, which are increasingly decoupled and distributed, and enables support staff to use a "white-box" approach to monitoring and diagnosing application health, faults, or other internal conditions.

Every released software application, regardless of size or complexity, imposes a common requirement on the business that the application serves: it must be managed to ensure that the application provides its services correctly and reliably during its operational lifetime. Instrumentation plays a key role in application manageability, allowing a particular software or hardware element to publish — or be queried for — relevant information. Examples of common instrumentation mechanisms include performance counters, event logs, Windows 2000 Event Trace, and Windows Management Instrumentation (WMI). These mechanisms are often complementary, as in the example of querying an event log through a WMI provider.

Achieving consistent instrumentation across all enterprise applications is a difficult task. Today, enterprises that build applications on Microsoft platforms must instrument their applications by directly writing to event logs, performance counters, third-party instrumentation APIs, or their own common instrumentation wrappers and libraries. Implementing and supporting the various forms of instrumentation brings additional challenges, given the distributed nature of today's n-tier, Web-enabled applications.

Operations staff must be able to trace specific paths through the system, not just monitor individual events and event sources. Logically related events from physically different servers need to be correlated. The instrumentation itself must be suitable for a production application; instrumentation overhead must minimally affect application throughput. Finally, organizations must be able to leverage as many existing management tools and infrastructure as possible, to monitor and troubleshoot the enterprise applications they support.

Key features of this framework are:

- Unified programming model, suitable for both enterprise developers and system developers.
- Structured WMI event schema, which acts as a supportability contract between Development, Test, and Operations teams.
- Scriptable configuration layer, allowing operations teams to configure how events are raised or logged from an application.
- Support for raising or logging events through WMI, Windows Event Log, and Windows Event Tracing, a high-speed kernel-mode tracing system.
- Correlation of events to business processes or operations with Request Tracing, which allows operations staff to troubleshoot requests across a distributed application.

In Addition, the Adapter incorporates the following enhancements to EIF:

- The Level of Tracing and Eventing for the Adapter can be set by running a Wizard in Administration Microsoft Management Console (MMC) Snap-In. Levels that can currently be set is for Production, Testing/QA and Development.
- Tracing Sessions can be enabled or disabled from the MMC Snap-In.
- Trace Files can be viewed and exported from the MMC Snap-In.

Performance Counters

The Adapters feature an extensive set of Windows Performance Counters with which the Adapters can be monitored and Fine Tuned.

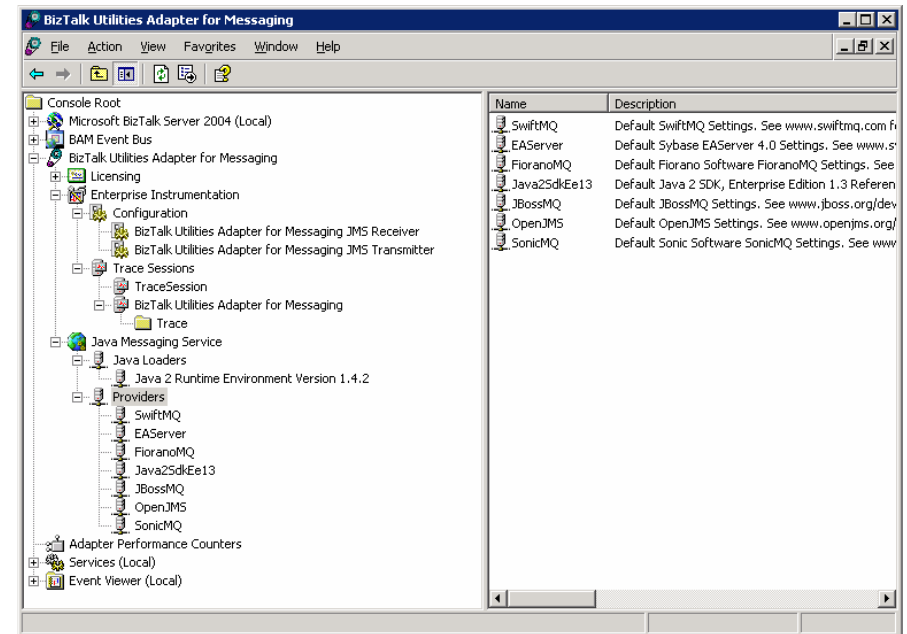
Exception Handlers

Send Adapters are equipped with a feature called Exception Handlers. It allows you to specify how a send Adapter should behave when a particular exception occurs in the transmission of a Message.

In addition a set of Regular Expressions called Matches can be specified that will perform a match on the Message of the Exception and perform a particular action like immediately suspending the Message.

Exception Handlers can be specified on a Port basis or Globally for all Send Ports.

Administration Snap-In



The Microsoft Management Console (MMC) Administration Snap-In is used to perform the following tasks:

- Updating of the License File for the Adapters.
- Configuration of Enterprise Instrumentation Settings for the Adapters.
- The Viewing of Trace Sessions and Files.
- Configuration of the preferred Java Virtual Machine for JMS.
- Creation, Deletion and Updating of JMS Provider Settings used within the JMS Adapters.
- Viewing of Performance Counters for the Adapters.

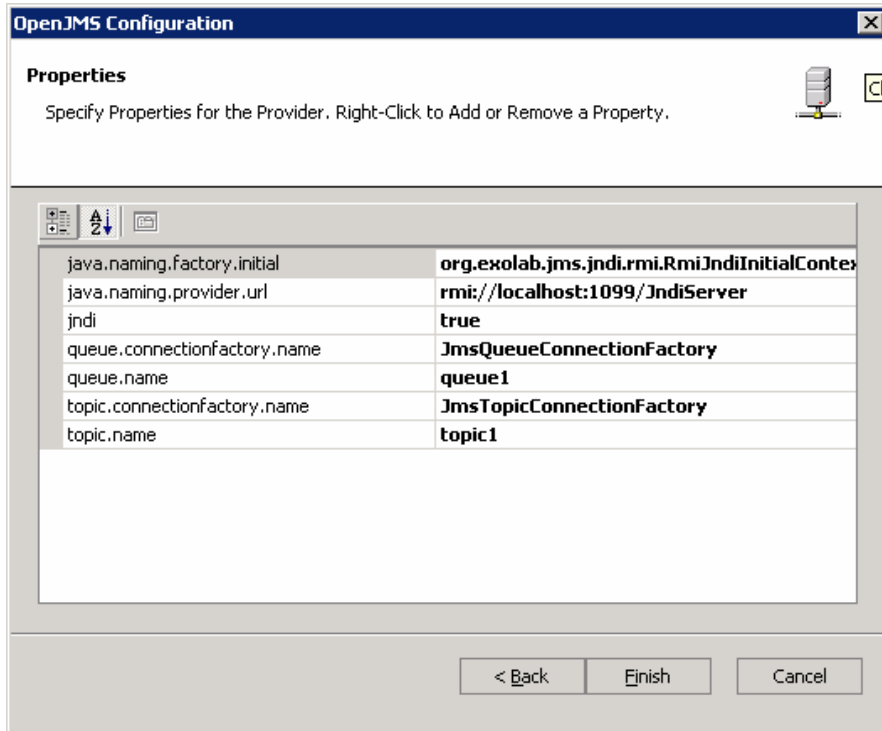
Archiving

As standard within all of the BizTalk Utilities Adapters, inbound messages can be archived to a File Location or Microsoft Message Queuing (MSMQ). Support for Archiving to BizTalk Server will be added in Future.

Java Messaging Service (JMS) Support

JMS Providers

JMS Providers define the connection parameters for connecting to a specified JMS Queue or Topic.

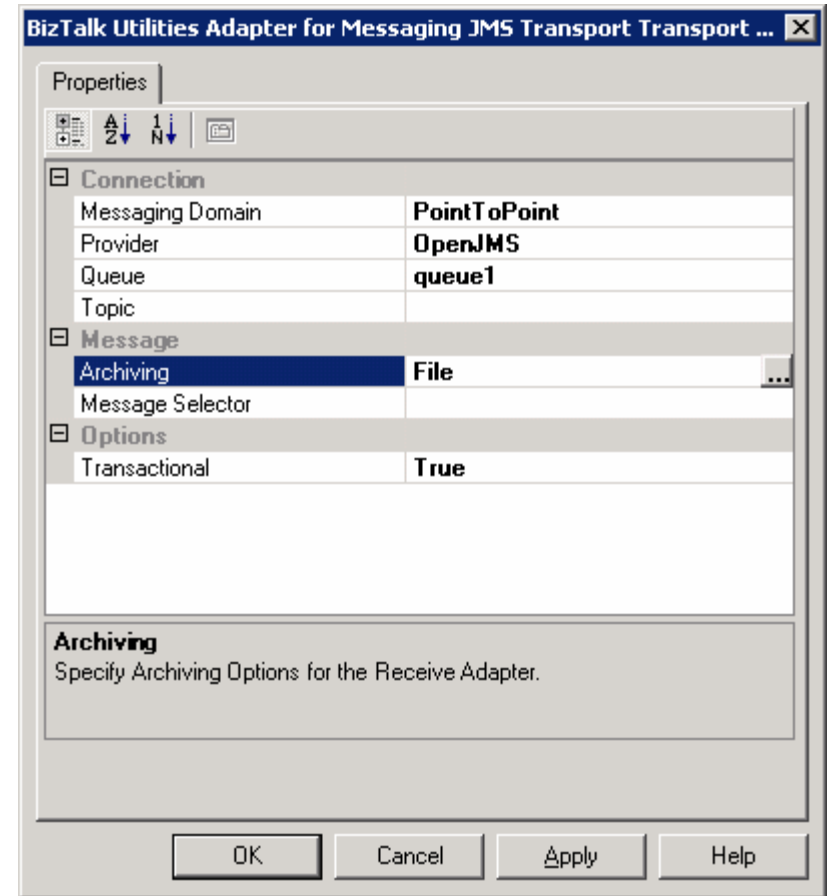


JMS Providers are defined within the BizTalk Utilities Adapter for Messaging MMC Snap-In. The Adapter installs a set of sample Providers for SwiftMQ, EAServer, FioranoMQ, JBossMQ, OpenJMS and SonicMQ.

With this version the User and Password can also be set.

JMS Receiver

The JMS Receiver is responsible for receiving Messages from either Queues or Topics and submitting them to BizTalk Server.



The following Properties can be set within the Receiver:

- **Messaging Domain**

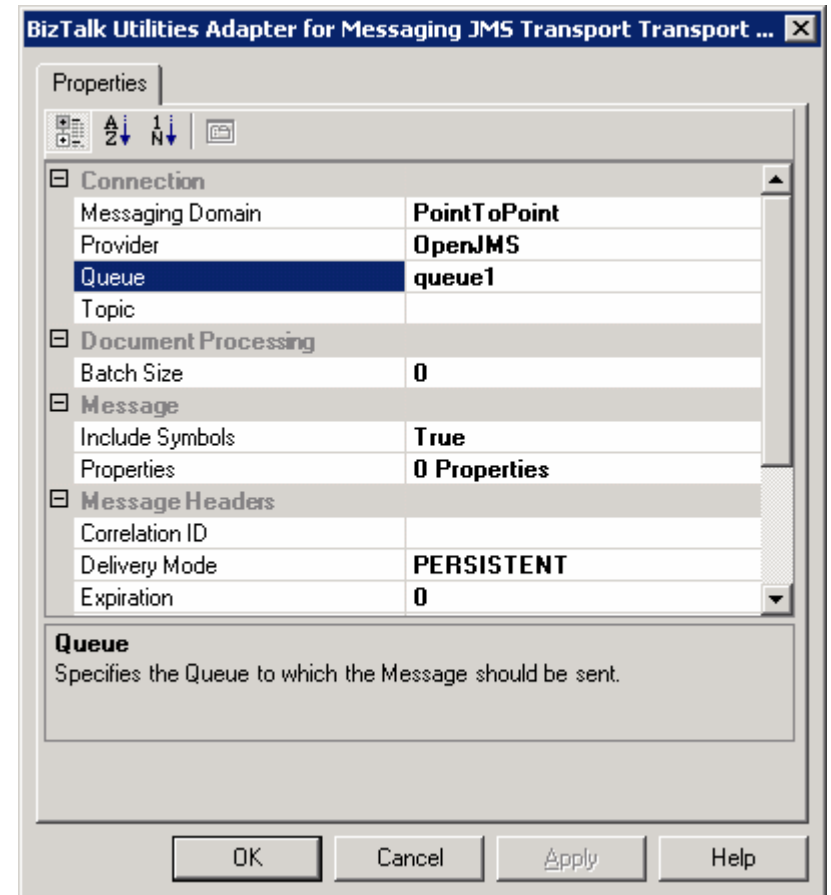
This can be either Point To Point or Publish-Subscribe.

- **Provider**
Here you select a JMS Provider as defined in the Administration Snap-In.
- **Queue**
Specify the name of the JMS Queue from which you want to receive Messages.
- **Topic**
Specify the JMS Topic from which you want to retrieve Messages.
- **Archiving**
Specify how you want to Archive Messages received.
- **Message Selector**
Specify Filter Criteria for the Message.
- **Transactional**
Specify if you want to retrieve Messages within a Transaction.

JMS Transmitter

The JMS Transmitter is responsible for sending Messages to JMS Queues and Topics. It can be defined statically or invoked dynamically by setting the Uri of a Port within your Orchestration to `jms://<Provider>/<MessagingDomain>/<Queue/Topic>`.

When defining it dynamically other properties can also be set.



The following Properties can bet set within the Transmitter:

- **Messaging Domain**
This can be either Point To Point or Publish-Subscribe.
- **Provider**
Here you select a JMS Provider as defined in the Administration Snap-In.
- **Queue**

- Specify the name of the JMS Queue to which you want to send Messages.
- **Topic**
Specify the JMS Topic to which you want to send Messages.
- **Batch Size**
Specify the Size of Message Batches. A Value of 0 switches Batching off.
- **Include Symbols**
When this option is selected Message Context Properties are sent as Message Properties.
- **Properties**
Define any other Message Properties.
- **Correlation ID**
Specify the JMS Correlation ID Property.
- **Delivery Mode**
Specify the JMS Delivery Mode Property.
- **Expiration**
Specify the JMS Expiration Property.
- **Message ID**
Specify the JMS Message ID Property.
- **Priority**
Specify the JMS Priority Property.
- **Redelivered**
Specify the JMS Redelivered Property.
- **Type**
Specify the JMS Type Property.
- **Transactional**
Specify if Messages should be sent within a Transaction.

MQSeries Support

MQSeries support provides an MQSeries Receiver and Transmitter. Standard features include the following:

- Access to local or remote Managers and Queues. The Adapter only requires that an MQSeries Client Version 5.2 or upwards be installed.
- Secure Socket Layer Support.

MQSeries Receiver

The MQSeries Receiver is used to retrieve Messages from MQSeries Queues and submit them to BizTalk Server. All Message Properties are promoted as context properties when a message is submitted to BizTalk Server.

The following Properties can be set within the Receiver:

The screenshot shows a dialog box titled "BizTalk Utilities Adapter for Messaging MQSeries Transport Trans...". The "Properties" tab is active, displaying a tree view of configuration sections:

- Destination**
 - Queue: test
 - Queue Manager: QM_server
- Document Processing**
 - Batch Size: 10
 - Interval: 0 Days, 0 Hours, 2 Minutes, 0
- Environment**
 - Channel: (empty)
 - Host: (empty)
 - Port: 0
- Message**
 - Archiving: (empty)
- Options**
 - Backout Count: 4

At the bottom, there is a "Cipher Suite" section with the text "Cipher Suite." and four buttons: OK, Cancel, Apply, and Help.

The following properties can be set within the Receiver:

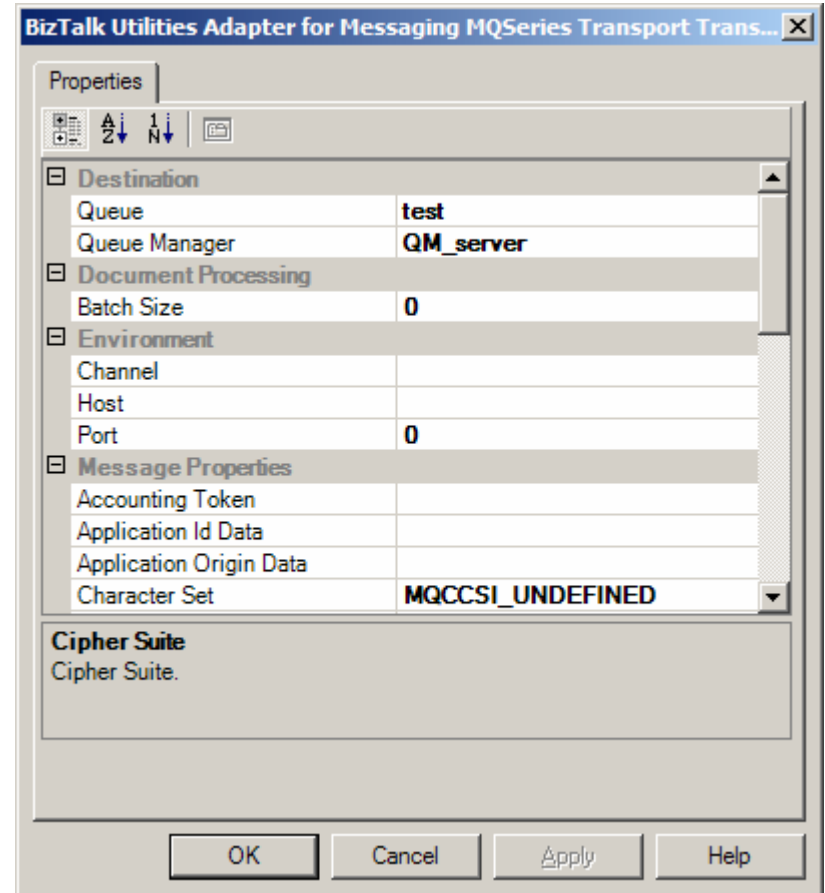
- **Queue**
Specifies the Queue that should be polled for Messages.
- **Queue Manager**
Specifies the Queue Manager where the Queue is located.
- **Batch Size**

- Specifies the number of Messages that should be retrieved per polling Interval.
- **Interval**
Specifies the Interval at which the MQSeries Receiver should poll the Queue for new Messages.
- **Channel**
Specifies the remote channel for accessing the Queue.
- **Host**
Specifies the Remote Host.
- **Port**
Specifies the remote Port.
- **Archiving**
Specifies Archiving options for Messages retrieved from the MQSeries Queue.
- **Backout Count**
If a message reaches the particular Backout count, the message will be moved to the Dead-Letter Queue for the Queue Manager. This is done avoid "poison" messages.
- **Transactional**
Specifies if Messages should be retrieved within a Transaction. A single Message is retrieved within a single Transaction.
- **Lock**
Specifies the Receive Location should be locked when processing a batch of messages..
- **Cipher Suite**
Specifies the SSL Cipher Suite to use when retrieving Messages.
- **Enabled**
Enables or Disables SSL.
- **Peer Name**
Specifies the SSL Peer Name.

MQSeries Transmitter

The MQSeries Transmitter is used to send message to MQSeries. The Transmitter can be configured in a static mode or dynamically by using by setting the Uri of a Port within your Orchestration to mqseries://<Queue Manager>/<Queue>.

All Message Properties can also be set using Context Properties.



The following Properties can be set within the Transmitter:

- **Queue**
Specifies the Queue that Messages should be sent to.
- **Queue Manager**
Specifies the Queue Manager where the Queue is located.
- **Batch Size**

- Specifies the number of Messages that should be submitted per Batch.
- **Channel**
Specifies the remote channel for accessing the Queue.
- **Host**
Specifies the Remote Host.
- **Port**
Specifies the remote Port.
- **Transactional**
Specifies if Messages should be submitted within a Transaction. A single Message is submitted within a single Transaction.
- **Cipher Suite**
Specifies the SSL Cipher Suite to use when retrieving Messages.
- **Enabled**
Enables or Disables SSL.
- **Peer Name**
Specifies the SSL Peer Name.

The following Message Properties can be set within the Transmitter:

- Accounting Token
- Application Id Data
- Application Origin Data
- Character Set
- Correlation Id
- Encoding
- Expiry
- Feedback
- Format
- Group Id
- Message Flags
- Message Id
- Message Sequence Number
- Message Type
- Persistence
- Priority
- Put Application Name
- Put Application Type
- Reply-To Queue Manager
- Reply-To Queue
- Report

About the Enabler Group

Established in April 2002, the Group aims to provide best of breed Services and Solutions to the BizTalk Marketplace.

Contact Details

Support	support@theenablergroup.com
Information	info@theenablergroup.com
Web Site	www.topxml.com/biztalkutilities/biztalk2004.asp
Telephone	+27 (11) 791 2797

The Enabler Group is a Microsoft Certified Partner.



