



Microsoft®
BizTalk® Server 2010

Microsoft BizTalk Server 2010 Operations Guide

Microsoft Corporation

Published: February, 2011

Summary

The BizTalk Server 2010 Operations Guide is a valuable resource for anyone involved in the implementation and administration of a BizTalk solution, particularly IT professionals. The guide provides detailed information for planning a BizTalk Server environment, as well as recommendations and best practices for configuring, testing, maintaining, and monitoring this environment. The Operations Guide contains checklists for all the key tasks to help BizTalk users.

Microsoft

Copyright

Information in this document, including URL and other Internet Web site references, is subject to change without notice. Unless otherwise noted, the companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted in examples herein are fictitious. No association with any real company, organization, product, domain name, e-mail address, logo, person, place, or event is intended or should be inferred. Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

© 2011 Microsoft Corporation. All rights reserved.

Microsoft, MS-DOS, Windows, Windows Server, Windows Vista, Active Directory, BizTalk, Excel, SharePoint, SQL Server, Visio, Visual C#, and Visual Studio are trademarks of the Microsoft group of companies. All other trademarks are property of their respective owners.

Contents

Operations Checklists.....	7
Operational Readiness Checklists	8
Checklist: Getting Started with BizTalk Server	9
Checklist: Configuring Windows Server.....	11
Checklist: Configuring Internet Information Services.....	17
Checklist: Configuring SQL Server.....	20
Checklist: Configuring BizTalk Server	34
Checklist: Providing High Availability with Fault Tolerance or Load Balancing	48
Checklist: Increasing Availability with Disaster Recovery.....	50
Checklist: Monitoring Operational Readiness.....	52
Checklist: Maintaining and Troubleshooting BizTalk Server Databases	54
Checklist: Testing Operational Readiness.....	72
Routine Maintenance Checklists.....	81
Checklist: Performing Daily Maintenance Checks	82
Checklist: Performing Weekly Maintenance Checks	83
Checklist: Performing Monthly Maintenance Checks	85
Routine Performance Checklists.....	88
Checklist: Performing Weekly Performance Checks	89
Checklist: Performing Monthly Performance Checks	91
Checklists for Other Important Tasks.....	93
Checklist: Deploying an Application.....	93
Checklist: Exporting Bindings from One Application to Another	96
Checklist: Updating an Assembly	96
Checklist: Updating an Application Using Side-by-Side Versioning	98
Checklist: Updating an Orchestration Using Side-by-Side Versioning	101
Checklist: Updating Artifacts in a BizTalk Application	103
Checklist: Installing and Configuring Certificates	103
Checklist: Monitoring BizTalk Server with Operations Manager 2007	104
Checklist: Monitoring SQL Servers.....	105
Checklist: Planning for Operations in a Secure Environment.....	105
Planning the Environment for BizTalk Server.....	112
BizTalk Server Roles and Responsibilities	115
Planning the BizTalk Server Tier.....	123
Planning for Sending and Receiving.....	123
Planning the BizTalk Solution	132
Planning for Single Sign-On	135
Planning for Tracking.....	136
Planning for Publishing Web Services.....	138
Planning for Consuming Web Services	142
Planning for Performance	146

Planning for High Availability and Disaster Recovery.....	147
Planning for Testing.....	148
BizTalk Server 64-Bit Support	149
Planning the Database Tier.....	149
Planning for Database Performance.....	150
Planning for Database Availability	151
Planning for Database Testing	152
Planning the Development, Testing, Staging, and Production Environments	153
Managing BizTalk Server	156
Managing Applications	157
Deploying an Application	158
Updating an Application	175
Managing Certificates	189
Best Practices for Managing Certificates.....	189
Known Issues with Certificates in BizTalk Server.....	191
Installing and Configuring Digital Certificates	194
Moving Databases.....	213
Moving BAM Databases	214
Moving Non-BAM Databases	233
Monitoring BizTalk Server	234
Monitoring the BizTalk Server Environment.....	234
Best Practices for Monitoring	238
Routine Monitoring Tasks	240
Using the Performance Analysis of Logs (PAL) Tool.....	242
Monitoring BizTalk Server with System Center Operations Manager 2007.....	286
Best Practices for Monitoring with Operations Manager 2007	286
Monitoring Throttling Using Performance Threshold Rules.....	287
Monitoring SQL Servers	291
Monitoring Applications and Host Instances.....	294
Maintaining BizTalk Server.....	297
Maintaining Reliability	297
Administrative Maintenance	301
Best Practices for Administrative Maintenance	302

Microsoft BizTalk Server 2010 Operations Guide

Welcome to the Microsoft® BizTalk® Server 2010 Operations Guide. We created this guide to be a valuable resource for anyone involved in the implementation and administration of a BizTalk solution, particularly IT professionals.

To download a copy of this guide in chm, pdf, or docx form, go to [Microsoft BizTalk Server 2010 Operations Guide](http://go.microsoft.com/fwlink/?LinkId=212652) (http://go.microsoft.com/fwlink/?LinkId=212652).

Which Versions of BizTalk Server Does the Guide Cover?

This guide caters to BizTalk Server 2010 and provides operational readiness information to help you jumpstart with your BizTalk Server setup.



Note

To see the Operations Guide for BizTalk Server 2006 R2 or BizTalk Server 2006, see <http://go.microsoft.com/fwlink/?LinkId=130458>.

Where Do I Start?

We organized the guide according to the functional aspects of planning, deploying, and managing a BizTalk Server installation. You can therefore read it according to these functional aspects. However, realizing that the checklists would be the most sought after information in the BizTalk Server 2010 Operations Guide, we have categorized all the checklists in the document in the following table for ease of accessibility.

Setting Up Your BizTalk Environment	Setting Up a High Availability and Disaster Recovery Environment
<ul style="list-style-type: none">• Checklist: Getting Started with BizTalk Server• Checklist: Configuring Windows Server• Checklist: Configuring Internet Information Services• Checklist: Configuring SQL Server• Checklist: Configuring BizTalk Server	<ul style="list-style-type: none">• Checklist: Providing High Availability with Fault Tolerance or Load Balancing• Checklist: Increasing Availability with Disaster Recovery
Monitoring, Testing, and Troubleshooting	Performance and Maintenance
<ul style="list-style-type: none">• Checklist: Monitoring Operational	Maintenance related checklists:

<p>Readiness</p> <ul style="list-style-type: none"> • Checklist: Maintaining and Troubleshooting BizTalk Server Databases • Checklist: Testing Operational Readiness • Checklist: Monitoring BizTalk Server with Operations Manager 2007 • Checklist: Monitoring SQL Servers 	<ul style="list-style-type: none"> • Checklist: Performing Daily Maintenance Checks • Checklist: Performing Weekly Maintenance Checks • Checklist: Performing Monthly Maintenance Checks <p>Performance related checklists:</p> <ul style="list-style-type: none"> • Checklist: Performing Weekly Performance Checks • Checklist: Performing Monthly Performance Checks
<p>Checklists for Other Important Tasks</p>	
<ul style="list-style-type: none"> • Checklist: Deploying an Application • Checklist: Exporting Bindings from One Application to Another • Checklist: Updating an Assembly • Checklist: Updating Artifacts in a BizTalk Application 	<ul style="list-style-type: none"> • Checklist: Updating an Application Using Side-by-Side Versioning • Checklist: Updating an Orchestration Using Side-by-Side Versioning • Checklist: Installing and Configuring Certificates • Checklist: Planning for Operations in a Secure Environment

If you are performing the following tasks, you can start with the related sections:

- **Evaluating operational readiness:** If you are focused on assessing and evaluating the operational readiness of a BizTalk Server deployment, then start by reading the [Operations Checklists](#) and [Increasing Availability for BizTalk Server](#) sections.
- **Becoming operationally ready:** To ensure that your BizTalk Server infrastructure and applications become operationally ready, refer to the [Planning the Environment for BizTalk Server](#) section.
- **Maintaining an operational environment:** Most of the topics in the operations guide assist you in maintaining an operational BizTalk Server environment. You will find best practices, key concepts, and procedures for maintaining an operational environment in the following sections: [Increasing Availability for BizTalk Server](#), [Monitoring BizTalk Server](#), and [Maintaining BizTalk Server](#).

What's in It?

Guidance based on real-world experience. The idea for the guide originated with Microsoft field representatives, partner organizations, and customers who plan, deploy, and maintain BizTalk Server installations. This group of IT professionals has accumulated extensive hands-on experience with a diverse range of BizTalk solutions. As they gained experience they created checklists, best practices, and presentations to guide future BizTalk Server operations. We collected and organized this information to create the guide.

Key portions of this guide are new; however, a considerable portion refers to BizTalk Server Help, white papers, Knowledge Base articles, and other sources. It has been carefully reviewed and vetted by experts from the community of BizTalk Server IT professionals and members of the product development team, whom we gratefully acknowledge at the end of this topic. We believe that the information presented here will help BizTalk Server users solve, and above all, avoid many of the common problems that can occur while deploying and maintaining a BizTalk Server installation.

Acknowledgments

We in the BizTalk Server User Education team gratefully acknowledge the outstanding contributions of the following individuals for providing both technical feedback as well as a good deal of content for the BizTalk Server Operations Guide:

- Paolo Salvatori, Tim Wieman

Operations Checklists

This section contains checklists for deploying and maintaining BizTalk Server.

Operational Readiness Checklists

The Operational Readiness checklists are arguably the heart of the operations guide. The Operational Readiness checklists are a product of many contributors, including the BizTalk Customer Advisory Team (previously known as the BizTalk Rangers), Microsoft Customer Service and Support (CSS), Microsoft Consulting Services (MCS), Microsoft partners, and customers. As such, the checklists represent some of the most important lessons learned and best practices compiled through experience with hundreds of BizTalk Server installations. The Operational Readiness checklists cover platform configuration (including the applications that BizTalk Server depends on for services), deployment, availability, monitoring, and testing.

Routine Maintenance Checklists

These checklists describe tasks and procedures that are the basis for a preventative maintenance program you should follow throughout the life of your BizTalk Server environment.

Routine Performance Checklists

These checklists describe tasks and procedures that are the basis for a preventative maintenance program to track the performance of your BizTalk Server solutions.

In This Section

- [Operational Readiness Checklists](#)
- [Routine Maintenance Checklists](#)

- [Routine Performance Checklists](#)
- [Checklists for Other Important Tasks](#)

Related Sections

[Managing BizTalk Server](#)

[Monitoring BizTalk Server](#)

[Maintaining BizTalk Server](#)

See Also

[Microsoft BizTalk Server 2010 Operations Guide](#)

Operational Readiness Checklists

The Operational Readiness checklists contain recommendations that you should consider and tasks that you should perform before deploying a BizTalk solution into production. These checklists include information for configuring prerequisite software, increasing availability, monitoring the BizTalk Server environment, and procedures for testing.

Because BizTalk Server maintains dependencies on so many other Microsoft technologies, you should complete the tasks for each dependent technology to help ensure that your production BizTalk Server environment runs smoothly.

Typical Prerequisite Software

Prerequisite software for the BizTalk Server application platform typically includes the following products:

- The Windows operating system
- SQL Server 2008 SP1 SP1 or SQL Server 2008 R2
- BizTalk Server
- Visual Studio 2010 (for development purposes, not at run time)

Additional Components

The BizTalk Server application platform may also require several of the following software components:

- Internet Information Services (IIS)
- Windows SharePoint® Services 2010
- SharePoint Foundation 2010
- Microsoft Office SharePoint Server 2007 Service Pack 1 (SP1) (MOSS)
- Windows SharePoint Services 3.0 with SP1 or SP2
- Microsoft Office Excel 2010 or 2007

 **Note**

BizTalk Server 2010 supports only the 32-bit version of Microsoft Office 2010.

- SQL Server 2005 Notification Services
- SQLXML 4.0 with Service Pack 1
- .NET Framework 1.0
- .NET Framework 2.0
- .NET Framework 3.0
- Microsoft .NET Framework 4 and .Net Framework 3.5 with Service Pack 1 (SP1)
- Non-Microsoft components to enable functionality for certain BizTalk Server adapters.

For more information about the dependency software that is required for specific features of the BizTalk application platform for various Windows operating system versions, see the "Feature Dependency Matrix" section in the BizTalk Server 2010 installation and upgrade guide for the specific Windows operating system version. The BizTalk Server 2010 installation and upgrade guides are available at <http://go.microsoft.com/fwlink/?LinkID=152913>.

In This Section

- [Checklist: Getting Started with BizTalk Server](#)
- [Checklist: Configuring Windows Server](#)
- [Checklist: Configuring Internet Information Services](#)
- [Checklist: Configuring SQL Server](#)
- [Checklist: Configuring BizTalk Server](#)
- [Checklist: Providing High Availability with Fault Tolerance or Load Balancing](#)
- [Checklist: Increasing Availability with Disaster Recovery](#)
- [Checklist: Monitoring Operational Readiness](#)
- [Checklist: Maintaining and Troubleshooting BizTalk Server Databases](#)
- [Checklist: Testing Operational Readiness](#)

Checklist: Getting Started with BizTalk Server

If you are completely new to BizTalk Server, you should follow the checklist in this topic to get to speed with BizTalk Server.

Steps	Reference
Learn about BizTalk Server	
Learn about the skill set required for using BizTalk Server.	See Prerequisite Skills and Knowledge (http://go.microsoft.com/fwlink/?LinkId=155861).
Understand and read about BizTalk Server 2010.	<ul style="list-style-type: none"> • See Introducing BizTalk Server 2010 (http://go.microsoft.com/fwlink/?LinkId=155862). • See Microsoft BizTalk Server 2010

Steps	Reference
	Technical Overview (http://go.microsoft.com/fwlink/?LinkId=202654).
Look at BizTalk Server posters to understand the BizTalk Server architecture.	See BizTalk Server Posters (http://go.microsoft.com/fwlink/?LinkId=155865).
Understand how to design a BizTalk Server system keeping in mind various aspects such as security, performance, availability, and virtualization.	See Designing the System Architectures for BizTalk Server (http://go.microsoft.com/fwlink/?LinkId=155866).
Look at some sample BizTalk Server deployment architectures for reference.	<ul style="list-style-type: none"> • See Sample BizTalk Server Architectures (http://go.microsoft.com/fwlink/?LinkId=155867) • See Sample Architectures for Small & Medium-Sized Companies (http://go.microsoft.com/fwlink/?LinkId=155868).
Read through the scenarios in which you can use BizTalk Server 2010 in your organization.	See Scenarios for Business Solutions (http://go.microsoft.com/fwlink/?LinkId=155869).
Read through BizTalk Server case studies to understand how organizations deploy BizTalk Server to meet their business goals.	See Security Case Studies for Small & Medium-Sized Companies (http://go.microsoft.com/fwlink/?LinkId=155870).
Look at the BizTalk Server Developer Center to access various BizTalk Server resources such as documentation, blogs, videos, webcasts, and announcements.	See BizTalk Server Developer Center (http://go.microsoft.com/fwlink/?LinkId=158758).
Start using BizTalk Server	
Install and configure BizTalk Server 2010. As part of BizTalk Server installation and configuration, you must also configure Windows Server and SQL Server for optimum BizTalk Server performance. If you are going to publish BizTalk Server Web services and WCF services, you must also configure the Internet Information Services (IIS).	<ul style="list-style-type: none"> • See BizTalk Server 2010 Installation Guides (http://go.microsoft.com/fwlink/?LinkId=183138). • See Checklist: Configuring Windows Server • See Checklist: Configuring Internet Information Services • See Checklist: Configuring SQL Server • See Checklist: Configuring BizTalk Server
Look at videos of BizTalk Server to learn how to perform key tasks using BizTalk Server.	See BizTalk Server Videos (http://go.microsoft.com/fwlink/?LinkId=155889).

Steps	Reference
Work with BizTalk Server tutorials to understand how the key features in BizTalk Server can be used together to create a BizTalk Server solution.	See BizTalk Server Tutorials (http://go.microsoft.com/fwlink/?LinkId=155890).
Download and use the BizTalk Server samples to see how to create a BizTalk Server solution using the key features.	See BizTalk Server 2010 SDK Samples (http://go.microsoft.com/fwlink/?LinkId=202655).
Refer to the BizTalk Server 2010 Help and other white papers to know how to perform specific tasks in BizTalk Server.	<ul style="list-style-type: none"> • Online Help – Microsoft BizTalk Server 2010 Help (http://go.microsoft.com/fwlink/?LinkId=155892). • Downloadable Help – BizTalk Server 2010 Documentation (http://go.microsoft.com/fwlink/?LinkId=202656).
Plug into the BizTalk Server community	
Get support for BizTalk Server.	See BizTalk Server Support (http://go.microsoft.com/fwlink/?LinkId=156438).
Participate in BizTalk-focused newsgroups.	See Newsgroups (http://go.microsoft.com/fwlink/?LinkId=156439).
Read through BizTalk-related blogs.	See BizTalk Server Blogs (http://go.microsoft.com/fwlink/?LinkId=49361).
Ask questions on BizTalk-related forums.	See BizTalk Server Forums (http://go.microsoft.com/fwlink/?LinkId=156441).
Latest about BizTalk Server	
Keep yourself updated with the latest on BizTalk Server.	See Microsoft BizTalk Server (http://go.microsoft.com/fwlink/?LinkId=156443).

See Also

[Operational Readiness Checklists](#)

Checklist: Configuring Windows Server

This topic lists steps that you should follow when preparing Windows Server for use in a production BizTalk Server environment.

Configure Windows Server

Steps	Reference
Configure MSDTC for BizTalk Server.	<ul style="list-style-type: none"> • See How to Enable MSDTC on the BizTalk Server (http://go.microsoft.com/fwlink/?LinkId=153236). • See Troubleshooting Problems with MSDTC (http://go.microsoft.com/fwlink/?LinkId=153237).
Configure firewall(s) for BizTalk Server.  Note This step is only required if one or more firewalls are in place in your BizTalk Server environment.	<ul style="list-style-type: none"> • See Required Ports for BizTalk Server (http://go.microsoft.com/fwlink/?LinkId=153238). • See Microsoft Knowledge Base article 154596, "How to configure RPC dynamic port allocation to work with firewalls" (http://go.microsoft.com/fwlink/?LinkId=153239).
Turn off hyperthreading on all computers running BizTalk Server and SQL Server in the BizTalk Server environment.	<ul style="list-style-type: none"> • It is critical that hyperthreading be turned off for computers running BizTalk Server. This is a BIOS setting, typically found in the Processor settings of the BIOS setup. Hyperthreading makes the server appear to have more processors/processor cores than it actually does; however, hyperthreaded processors typically provide between 20% and 30% of the performance of a physical processor/processor core. When BizTalk Server counts the number of processors to adjust its self-tuning algorithms, the hyperthreaded processors cause these adjustments to be skewed, which is detrimental to overall performance. • Hyperthreading should be turned off for SQL Server computers because applications that can cause high levels of contention (such as BizTalk Server) can cause decreased performance in a hyper-threaded environment on a SQL Server computer.
Ensure Windows Server processor scheduling is set to "Background services".	Ensuring this configuration option is set on all computers running Windows Server in your

Steps	Reference
	<p>environment will improve the overall system performance. Follow these steps to ensure Windows Server is configured to favor background services:</p> <ol style="list-style-type: none"> 1. Click Start, click Run, and then type sysdm.cpl in the Run box. 2. In the System Properties dialog box, click the Advanced tab, and then click Settings under Performance. 3. In the Performance Options dialog box, click the Advanced tab, make sure the Background services option is selected under Processor scheduling, click OK, and then click OK again to close System Properties dialog box.
<p>Place the Windows paging file on a separate local physical drive.</p>	<p>Moving the paging file to a separate physical volume other than the operating system on a computer running Windows Server improves performance by reducing disk contention. Follow these steps to move the paging file to a separate physical volume other than the operating system:</p> <ol style="list-style-type: none"> 1. Click Start, click Run, and then type sysdm.cpl in the Open box. 2. Click the Advanced tab, and then click Settings under Performance. 3. Click the Advanced tab, click Change under Virtual memory, specify options for the paging file, click OK and then click OK again to close System Properties. <p> Note You must restart the computer for the new settings to take effect.</p>
<ul style="list-style-type: none"> • Defragment all disks (local and SAN/NAS) on a regular basis by scheduling off-hours disk defragmentation. • Defragment the Windows paging file and pre-allocate the Master File Tables of each disk in the BizTalk Server environment to boost overall system performance. 	<p>Use the PageDefrag utility (http://go.microsoft.com/fwlink/?LinkId=108976) to defragment the Windows paging file and pre-allocate the Master File Tables.</p>

Steps	Reference
If antivirus software is installed on the computer running SQL Server, disable real-time scanning of the data and transaction files (.mdf, .ndf, .ldf, .mdb).	Real-time scanning of the SQL Server data and transaction files can increase disk I/O contention and reduce SQL Server performance.
If antivirus software is installed on the computer running BizTalk Server, disable real-time scanning of non-executable file types referenced by any BizTalk Server receive locations (usually .XML, but can also be .csv, .txt, etc.).	Real-time scanning of non-executable files that are referenced by BizTalk Server receive locations can increase I/O contention on these files and reduce BizTalk Server performance.
If intrusion detection software is installed, disable network scanning between computers running BizTalk Server and external data repositories (SQL Server) or messaging services (such as Message Queuing and WebSphere MQSeries).	Intrusion detection software can slow down or even prevent valid communications over the network.
Network card (NIC) drivers on all computers in the BizTalk Server environment should be tuned for performance.	Adjust the network device drivers to maximize the amount of memory available for packet buffering, both incoming and outgoing. Also maximize buffer counts, especially transmit buffers and coalesce buffers. The default values for these parameters, and whether they are even provided, vary between manufacturers and driver versions. The goal is to maximize the work done by the network interface card hardware, and to allow the greatest possible buffer space for network operations to mitigate network traffic bursts and associated congestion.
Set the network cards to a fixed speed and duplex	Use a fixed speed and duplex (1 Gigabit or higher with full duplex) for the network connections on the BizTalk and SQL servers. This will ensure that the network interface does not auto-negotiate a lower speed or duplex setting, which has been a problem with some enterprise switches in the past. Also, in high-volume environments, it is advisable to have Gigabit networks.
Stop or disable any Windows services that are	Running unnecessary services on a production

Steps	Reference
not strictly necessary (such as Print Spooler and Indexing Service) on all computers in the BizTalk Server environment.	server uses system resources which could otherwise be used by BizTalk Server or SQL Server.

In This Section

- [Installing COM+ Hotfix Rollup Packages](#)

See Also

[Operations Checklists](#)

Installing COM+ Hotfix Rollup Packages



Note

This topic is applicable only for Windows Server 2003.

You should install the last version of the Windows Server 2003 COM+ hotfix rollup package and latest version of the Distributed Transaction Coordinator (DTC) rollup package. This is because the packages include many performance and stability fixes.

You should install these rollups on all computers running BizTalk Server and on all computers running SQL Server. The rollups are especially important for your BizTalk solution to perform well in high-throughput scenarios.

The DTC issues that have been fixed are as follows:

- Unexpected DTC shutdown
- Memory fragmentation issues
- DTC may stop responding under load
- DTC may leak memory or stop responding when used with BizTalk Server
- Performance of DTC under load has been improved

Installing the COM+ Rollup Package

COM+ is no longer releasing rollup packages. Follow this information to install the last COM+ package:

- The final version of the COM+ rollup is the “Windows Server 2003 Post-Service Pack 2 COM+ 1.5 Hotfix Rollup Package 12”. This hotfix will install on any version of Windows Server 2003, even those without a service pack installed. More information about this hotfix can be found in Microsoft Knowledge Base article 934016, ["Availability of Windows Server 2003 Post-Service Pack 2 COM+ 1.5 Hotfix Rollup Package 12"](#) (<http://go.microsoft.com/fwlink/?LinkId=158756>).

Installing the DTC Rollup Package

DTC will be releasing rollup packages independent of COM+. Follow this information to install the latest DTC package:

- As of this writing, the latest DTC hotfix rollup is “Package 16”. More information about this hotfix can be found in Microsoft Knowledge Base article 958013, ["List of the MS DTC issues"](#)

[that are fixed in Windows Server 2003 MS DTC Hotfix Rollup Package 16"](http://support.microsoft.com/kb/979919)
(<http://support.microsoft.com/kb/979919>).

The KB article about the latest DTC hotfix rollup package can be found by searching <http://support.microsoft.com> for the phrase (including the quotes):

"MS DTC Hotfix Rollup Package"

The following query does this search for you. Choose the latest one:

[http://support.microsoft.com/search/default.aspx?query="MS+DTC+Hotfix+Rollup+Package"](http://support.microsoft.com/search/default.aspx?query=)

Disabling Windows Server 2003 SP1 and SP2 Denial of Service Checking

Note

This topic is applicable only for Windows Server 2003.

You should disable the Windows Server 2003 Service Pack 1 and Service Pack 2 denial of service checking. This is because under certain high-load scenarios, Windows Server 2003 SP1 and SP2 denial of service checking may incorrectly identify valid TCP/IP connections as a denial of service attack.

Important

You should disable this feature only in an intranet scenario when you are sure you will not suffer from actual denial of service attacks.

How Denial of Service Can Affect TCP/IP Connections

Windows Server 2003 SP1 and SP2 implement a security feature that reduces the size of the queue for concurrent TCP/IP connections to the server. This feature helps prevent denial of service attacks. Under heavy load conditions, the TCP/IP protocol in Windows Server 2003 SP1 or later may incorrectly identify valid TCP/IP connections as a denial of service attack. This may occur when BizTalk Server is under heavy load.

Modifying the Registry Entry

For more information, see Microsoft Knowledge Base article 899599, "[A BizTalk Server Host instance fails, and a 'General Network' error is written to the Application log when the BizTalk Server-based server processes a high volume of documents"](http://go.microsoft.com/fwlink/?LinkId=158860)

(<http://go.microsoft.com/fwlink/?LinkId=158860>). Follow the instructions in this article to create the **SynAttackProtect** registry entry on computers running SQL Server that host BizTalk Server databases and on any computers running BizTalk Server that are running Windows Server 2003 SP1 or later.

Tuning Registry Settings that Govern the Level of Denial of Service Attack Protection

In certain scenarios you may want to maintain denial of service protection but reduce how aggressively the denial of service functionality is applied. It is possible to tune the default behavior of the denial of service protection feature by following these steps:

1. Ensure that the **SynAttackProtect** registry entry is set to a REG_DWORD value of 1 as described at <http://go.microsoft.com/fwlink/?LinkId=111477>.
2. Configure the **TcpMaxHalfOpen** registry entry as described at <http://go.microsoft.com/fwlink/?LinkId=111478>.

3. Configure the **TcpMaxHalfOpenRetried** registry entry as described at <http://go.microsoft.com/fwlink/?LinkId=111479>.

See Also

[Operational Readiness Checklists](#)

Monitoring and Reducing DTC Log File Disk I/O Contention

The Distributed Transaction Coordinator (DTC) log file can become a disk I/O bottleneck in transaction-intensive environments. This is especially true when using adapters that support transactions, such as SQL Server, MSMQ, or MQSeries, or in a multi-MessageBox environment. Transactional adapters use DTC transactions, and multi-MessageBox environments make extensive use of DTC transactions.

Monitoring Usage in Clustered and Non-Clustered Environments

To ensure that the DTC log file does not become a disk I/O bottleneck, you should monitor the disk I/O usage for the disk where the DTC log file resides on the SQL Server database server. In an environment where SQL Server is clustered, this is not as much of a concern because the log file will already be on a shared drive, which will likely be a fast SAN drive with multiple spindles. You should nevertheless still monitor the disk I/O usage since it can become a bottleneck in non-clustered environments or when the DTC log file is on a shared disk with other disk-intensive files.

Troubleshooting DTC

For information about troubleshooting DTC, see "Troubleshooting Problems with MSDTC" in BizTalk Server 2010 Help at <http://go.microsoft.com/fwlink/?LinkId=153237>.

See Also

[Checklist: Configuring Windows Server](#)

Checklist: Configuring Internet Information Services

This topic lists steps that should be followed when preparing Internet Information Services (IIS) for use in a production BizTalk Server environment.

Steps	Reference
Set up IIS to publish BizTalk Server Web services and WCF services	<ul style="list-style-type: none"> • See "Enabling Web Services" (http://go.microsoft.com/fwlink/?LinkId=153335) in the BizTalk Server documentation. • See "Configuring IIS for the Isolated WCF Receive Adapters" (http://go.microsoft.com/fwlink/?LinkId=202825) in the BizTalk Server documentation.
Verify that BizTalk Server Web services and WCF services are working correctly.	<ul style="list-style-type: none"> • See "Testing Published Web Services" (http://go.microsoft.com/fwlink/?LinkId=153336) in the BizTalk Server documentation. • See "How to Create a .NET Application to Test a

Steps	Reference
	<p>WCF Service Published with the BizTalk WCF Service Publishing Wizard (http://go.microsoft.com/fwlink/?LinkId=202830) in the BizTalk Server documentation.</p>
<p>Lock down BizTalk Server Web services:</p> <ul style="list-style-type: none"> • Turn off the Debug switch in the web.config or machine.config file. • Configure so that POST is the only verb allowed. 	<ul style="list-style-type: none"> • See Microsoft Knowledge Base Article 815145, "HOW TO: Lock Down an ASP.NET Web Application or Web Service" (http://go.microsoft.com/fwlink/?LinkId=153337). • See "ASP.NET Edit Rule Dialog Box" (http://go.microsoft.com/fwlink/?LinkId=64566) in the .NET Framework 2.0 documentation.
<p>Configure load balancing by using NLB (or other load balancer) to balance load across BizTalk Server Web services and WCF services.</p>	<ul style="list-style-type: none"> • For Windows Server 2008: See "Network Load Balancing Deployment Guide" (http://go.microsoft.com/fwlink/?LinkId=153139). • For Windows Server 2003: See "Network Load Balancing: Configuration Best Practices for Windows 2000 and Windows Server 2003" (http://go.microsoft.com/fwlink/?LinkId=69529).
<p>Change IIS and ASP.NET settings for tuning Web services.</p> <p> Note ASP.NET 2.0 includes auto-tuning, so modifying these settings should not be needed for the web.config file of ASP.NET 2.0 Web sites where autoConfig is enabled in the <processModel> section. "autoConfig=true" is the default setting.</p>	<p>Review the "ASP.NET settings that can impact HTTP or SOAP Adapter performance" section of the topic "Configuration Parameters that Affect Adapter Performance" (http://go.microsoft.com/fwlink/?LinkId=153338) in the BizTalk Server documentation.</p>
<p>Implement an approach for publishing BizTalk Server Web services and WCF services.</p>	<ul style="list-style-type: none"> • See Publishing Internet-facing Web Services and WCF Services. • See "Publishing WCF Services" (http://go.microsoft.com/fwlink/?LinkId=202827) in the BizTalk Server documentation.
<p>Follow best practices for optimizing IIS performance.</p>	<p>See "Top Ten Ways To Pump Up IIS Performance" (http://go.microsoft.com/fwlink/?LinkId=109107).</p>
<p>Follow best practices for writing high performance web applications for IIS.</p>	<p>See "10 Tips for Writing High-Performance Web Applications"</p>

Steps	Reference
	(http://go.microsoft.com/fwlink/?LinkId=98290).

In This Section

- [Publishing Internet-facing Web Services and WCF Services](#)

Publishing Internet-facing Web Services and WCF Services

You can use multiple approaches for publishing BizTalk Server Web services and WCF services to the Internet:

- Use reverse proxy rules in a perimeter network (also known as DMZ, demilitarized zone, and screened subnet).
- Put the computers running BizTalk Server that publish the Web services or WCF services into the perimeter network domain.
- Use BizTalk Server 2010 cloud enabler functionality to publish the Web services or WCF services as an Azure AppFabric Service Bus relay endpoint.

Using a Reverse Proxy

This has been the traditional approach for publishing BizTalk Server Web services and WCF services. Using reverse proxy rules in the perimeter network obviates the need to have BizTalk servers located in the perimeter network. The reverse proxy rules simply forward the HTTP and SOAP requests from the perimeter network to the computers running BizTalk Server in the intranet domain.

For more information about using a reverse proxy, see the following topics in BizTalk Server 2010 Help:

- ["Sample Architecture: HTTP and SOAP Adapters"](#) (http://go.microsoft.com/fwlink/?LinkId=153339).
- ["Sample TMA: HTTP and SOAP Adapters"](#) (http://go.microsoft.com/fwlink/?LinkId=153340).
- ["Large Distributed Architecture"](#) (http://go.microsoft.com/fwlink/?LinkId=153341).

Using Computers Running BizTalk Server in the Perimeter Network

This is not the preferred approach for publishing BizTalk Server Web services or WCF services to the Internet because it requires computers running BizTalk Server to be located in the perimeter network. However, when a reverse proxy is not available in the perimeter network, you can use this approach.

This approach requires the perimeter network domain to enlist in a one-way trust with the intranet domain (but the intranet domain does not trust the perimeter network domain). The IIS application pools that host the Web services or WCF services in the perimeter network domain must be running under an intranet domain account that is in the "BizTalk Isolated Host Users" domain group. This gives the application pool the required rights to publish messages to the BizTalk Server MessageBox database.

You must open specific ports in the firewall to accommodate this. For more information about the required ports, see ["Ports for the Receive and Send Servers"](#) (<http://go.microsoft.com/fwlink/?LinkId=153342>) in the BizTalk Server documentation.

Exposing BizTalk Applications on the Cloud using AppFabric Connect for Services

See the article [Exposing BizTalk Applications on the Cloud using AppFabric Connect for Services](#) (<http://go.microsoft.com/fwlink/?LinkID=204700>) for more information about expose BizTalk Applications as WCF Services on the cloud.

See Also

[Planning for Publishing Web Services](#)

Checklist: Configuring SQL Server

This topic lists steps that you should follow when preparing SQL Server 2008 SP1 or SQL Server 2005 for use in a BizTalk Server production environment.

- [Checklist: Configuring SQL Server](#)
- [Performing SQL Server Maintenance Procedures](#)
- [Backing Up the BizTalk Server Databases](#)
- [Using SQL Server Log Shipping For Disaster Recovery](#)
- [Monitoring BizTalk Server SQL Agent Jobs](#)
- [Purging and Archiving Tracking Data](#)

Configuring SQL Server

Steps	Reference
Monitor and reduce BizTalk Server database files disk I/O contention.	<ul style="list-style-type: none"> • We recommend that you proactively monitor the disk I/O usage for the disks that house the data and transaction log files. • We recommend that the data files and transaction log files for each of these be placed on dedicated drives to reduce the likelihood of disk I/O contention becoming a problem. • You can reduce disk I/O contention by separating the MessageBox and Tracking (DTA) databases, and by separating the database files and transaction log files on different physical disks. <p>For more information, see Monitoring and Reducing Database I/O Contention</p>
Ensure SQL Server is configured on properly-aligned disk partitions	Properly aligned disk partitions could result in significant decrease in latency thereby improving the SQL Server performance, which in turn enhances BizTalk Server performance. On the contrary, nonaligned disk

Steps	Reference
	<p>partitions can negatively affect I/O performance, thereby degrading the SQL Server and BizTalk Server performance.</p> <p>For more information about how properly aligned disk partitions can positively affect performance, see "Disk Partition Alignment Best Practices for SQL Server" (http://go.microsoft.com/fwlink/?LinkId=154073).</p>
<p>Keep the events you monitor with the SQL Server Profiler</p>	<p>Use SQL Server Profiler to monitor only the events in which you are interested. If traces become too large, you can filter them based on the information you want, so that only a subset of the event data is collected. Monitoring too many events adds overhead to the server and the monitoring process, and can cause the trace file or trace table to grow very large, especially when the monitoring process takes place over a long period of time.</p>
<p>Monitor and reduce DTC log file disk I/O contention.</p>	<p>Monitoring and Reducing DTC Log File Disk I/O Contention</p>
<p>Provide high availability for the SQL Server databases.</p>	<p>Planning for Database Availability</p>
<p>Review active/active SQL Server cluster configuration for failover scenarios.</p>	<p>Reviewing and Testing SQL Server Cluster Configuration for Failover Scenarios</p>
<p>Use default configuration settings for:</p> <ul style="list-style-type: none"> • Max Degree of Parallelism (MDOP). • SQL Server statistics on the BizTalk Server MessageBox database. • SQL Server database index rebuilds and defragmentation. 	<p>SQL Server Settings That Should Not Be Changed</p>
<p>Enable trace flag 1118 (TF1118) as a startup parameter for all instances of SQL Server.</p>	<p>Implementing TF1118 helps reduce contention across the SQL Server instances by removing almost all single page allocations. For more information see Microsoft Knowledge Base article 328551, "Concurrency enhancements for the tempdb database" (http://go.microsoft.com/fwlink/?LinkId=153694).</p> <p> Note</p>

Steps	Reference
	<p>For more information about TF1118 see “Misconceptions around TF1118” (http://go.microsoft.com/fwlink/?LinkId=158271). Note that the content at this link is not owned by Microsoft and Microsoft does not guarantee the accuracy of the content.</p>
<p>Split the tempdb database into multiple data files of equal size on each SQL Server instance used by BizTalk Server.</p>	<p>Ensure that the data files used for the tempdb are of equal size. This is critical because the proportional fill algorithm used by SQL Server is based on the size of the data files. If data files are created with unequal sizes, the proportional fill algorithm will use the largest file more for Global Allocation Map (GAM) allocations rather than spreading the allocations between all the files, thereby defeating the purpose of creating multiple data files. As a general guideline, create one data file for each CPU on the server and then adjust the number of files up or down as necessary. Note that a dual-core CPU is considered to be two CPUs. In any event, the number of data files must not be greater than 8 no matter how many additional cores are available on the computer. For more information about tempdb files, see Optimizing tempdb Performance (http://go.microsoft.com/fwlink/?LinkId=158272) in the SQL Server documentation.</p>
<p>If you are running BizTalk Server and hosting your databases on computers running SQL Server with sufficient physical memory, configure the instances of SQL Server to use Address Windowing Extensions (AWE) memory.</p>	<p>On an x86 computer, SQL Server should be configured to use more than 2 GB of physical memory. SQL Server includes support for the use of Microsoft Windows Address Windowing Extensions (AWE) to address up to 32 GB of memory. With AWE, SQL Server can reserve memory that is not in use for other applications and the operating system. Each instance that uses this memory, however, must statically allocate the memory it needs. SQL Server can only use this AWE allocated memory for the data cache and not for executables, drivers, DLLs, and so forth. For more information, see:</p> <ul style="list-style-type: none"> • Microsoft Knowledge Base article 274750, "How to configure SQL Server to use more than 2 GB of physical memory" (http://go.microsoft.com/fwlink/?LinkId=153718). • Using AWE (http://go.microsoft.com/fwlink/?LinkId=153720) in

Steps	Reference
	<p>the SQL Server documentation.</p> <ul style="list-style-type: none"> • Enabling AWE Memory for SQL Server (http://go.microsoft.com/fwlink/?LinkId=158273) in the SQL Server documentation.
<p>Set the Minimum and Maximum Server memory to the same values on the SQL Server instance(s) that host the BizTalk Server databases.</p>	<p>The computers running SQL Server that host the BizTalk Server databases should be dedicated to running SQL Server. When the computers running SQL Server that host the BizTalk Server databases are dedicated to running SQL Server, we recommend that the 'min server memory' and 'max server memory' options on each SQL Server instance be set to specify the fixed amount of memory to allocate to SQL Server. In this case, you should set the 'min server memory' and 'max server memory' to the same value (equal to the maximum amount of physical memory that SQL Server will use). This will reduce overhead that would otherwise be used by SQL Server dynamically managing these values. Execute the following T-SQL commands on each computer running SQL Server to specify the fixed amount of memory to allocate to SQL Server:</p> <pre>sp_configure 'Max Server memory (MB)', (max size in MB) sp_configure 'Min Server memory (MB)', (min size in MB)</pre> <p>Before you set the amount of memory for SQL Server, determine the appropriate memory setting by subtracting the memory required for Windows Server from the total physical memory. This is the maximum amount of memory you can assign to SQL Server.</p> <p> Note</p> <p>If the computer(s) running SQL Server that host the BizTalk Server databases also host the Enterprise Single Sign-On master secret as described in the topic Clustering the Master Secret Server then you may need to adjust this value to ensure that there is sufficient memory available to run the Enterprise Single Sign-On Service.</p>
<p>Account for the size of BizTalk</p>	<ul style="list-style-type: none"> • When determining the size of messages in the

Steps	Reference
Tracking database	<p data-bbox="769 317 1365 415">BizTalk Tracking (DTA) database, add the average size of the message context to the message size if it is significant compared to the message size.</p> <ul data-bbox="724 428 1349 667" style="list-style-type: none"> <li data-bbox="724 428 1349 527">• To limit the size of messages in the BizTalk Tracking database, limit the number of properties that you promote. <li data-bbox="724 539 1349 667">• If the orchestration debugger option is enabled, take into account that the status of each shape in the orchestration is saved in the BizTalk Tracking database.

Performing SQL Server Maintenance Procedures

Steps	Reference
Define auto-growth settings for the BizTalk Server databases.	<ul data-bbox="821 919 1373 1444" style="list-style-type: none"> <li data-bbox="821 919 1373 1226">• Database auto-growth should be set to a fixed number of megabytes instead of a percentage, especially for the MessageBox and Tracking databases. Depending on your BizTalk Server application and throughput, the MessageBox and Tracking databases can get quite large. If auto-growth is set to a percentage, then the auto-growth can be substantial as well. <li data-bbox="821 1239 1373 1337">• Instant file initialization can greatly reduce the performance impact of a file growth operation. <li data-bbox="821 1350 1373 1444">• Ideally, the size of files supporting the file groups should be pre-allocated, and if possible, set to a static size. <p data-bbox="821 1457 1300 1528">For more information, see Defining Auto-Growth Settings for Databases.</p>
Back up the BizTalk Server databases	<ul data-bbox="821 1551 1373 1829" style="list-style-type: none"> <li data-bbox="821 1551 1373 1682">• We recommend running the BizTalk Server backup job to prevent the BizTalk Server database transaction logs from growing in an unbounded fashion. <li data-bbox="821 1694 1373 1793">• You should restore the entire BizTalk Server environment on a regular basis, and you should carefully document the process. <li data-bbox="821 1806 1373 1829">• We recommend that you archive old

Steps	Reference
	<p>backup files.</p> <p>For more information, see Backing Up Databases.</p>
<p>Monitor the BizTalk Server SQL Agent jobs.</p>	<p>Monitor the health of these jobs, and ensure that they are running without errors. For more information, see Monitoring SQL Server Agent Jobs.</p>
<p>Enable BizTalk Server tracking and archiving</p>	<p>The “DTA Purge and Archive” SQL Agent job archives and purges old data from the BizTalk Tracking database, thus keeping it from growing out of control. This is essential for a healthy BizTalk Server system. For more information, see Purging and Archiving Tracking Data.</p>

Backing Up the BizTalk Server Databases

Steps	Reference
<p>Verify that the Backup BizTalk Server SQL Agent job is configured.</p>	<ul style="list-style-type: none"> • See "How to Configure the Backup BizTalk Server Job" (http://go.microsoft.com/fwlink/?LinkId=153813). • See Backing Up Databases.
<p>Configure the Backup BizTalk Server SQL Agent job to delete backup files that are older than the number of days specified by the @DaysToKeep variable. If the backup files are not deleted they can grow unbounded over time which can fill up the disk(s) that contain the backup files and cause problems that relate to limited disk space.</p>	<p>See Microsoft Knowledge Base Article 982546, The "Backup BizTalk Server" job fails when backup files accumulate over time in Microsoft BizTalk Database Server (http://go.microsoft.com/fwlink/?LinkId=202858).</p>
<p>Verify that the Backup BizTalk Server SQL Agent job is enabled and running.</p>	<p>Monitoring SQL Server Agent Jobs</p>

Using SQL Server Log Shipping For Disaster Recovery

Steps	Reference
Verify that the disaster recovery servers have the capacity to handle production load.	See Using BizTalk Server Log Shipping for Disaster Recovery
Ensure that the specifics of your disaster recovery routine are well documented.	See Using BizTalk Server Log Shipping for Disaster Recovery
As part of regular testing, practice failover to the disaster recovery site, especially as new BizTalk applications are put in production.	See Using BizTalk Server Log Shipping for Disaster Recovery

Monitoring BizTalk Server SQL Agent Jobs

Steps	Reference
Verify that the SQL Server Agent service is running.	See Monitoring SQL Server Agent Jobs
Verify that the SQL Server Agent jobs installed by BizTalk Server are enabled and running successfully.	See Monitoring SQL Server Agent Jobs
Verify that the BizTalk Server SQL Agent jobs are completing in a timely manner.	See Monitoring SQL Server Agent Jobs

Purging and Archiving Tracking Data

Steps	Reference
Ensure that the SQL Agent job "DTA Purge and Archive" is properly configured, enabled, and successfully completing.	See "How to Configure the DTA Purge and Archive Job" (http://go.microsoft.com/fwlink/?LinkId=153814).
Ensure that the job is able to purge the tracking data as fast as the incoming tracking data is generated.	See "Measuring Maximum Sustainable Tracking Throughput" (http://go.microsoft.com/fwlink/?LinkId=153815).
Review the soft purge and hard purge parameters to ensure you are keeping data for the optimal length of time.	See "Archiving and Purging the BizTalk Tracking Database" (http://go.microsoft.com/fwlink/?LinkId=153816).
If you only need to purge the old data and do not need to archive it first, change the SQL Agent job to call the stored procedure "dtasp_PurgeTrackingDatabase."	See "How to Purge Data from the BizTalk Tracking Database" (http://go.microsoft.com/fwlink/?LinkId=153817).

In This Section

- [SQL Server Settings That Should Not Be Changed](#)
- [Monitoring and Reducing Database I/O Contention](#)
- [Reviewing and Testing SQL Server Cluster Configuration for Failover Scenarios](#)
- [Defining Auto-Growth Settings for Databases](#)
- [Backing Up Databases](#)
- [Using BizTalk Server Log Shipping for Disaster Recovery](#)
- [Monitoring SQL Server Agent Jobs](#)
- [Purging and Archiving Tracking Data](#)

SQL Server Settings That Should Not Be Changed

When setting up SQL Server during the operational readiness procedures for BizTalk Server, you should not make changes to the following settings.

SQL Server Max Degree of Parallelism

Max Degree of Parallelism (MDOP) is set to “1” during the configuration of BizTalk Server for the SQL Server instance(s) that host the BizTalk Server MessageBox database(s). This is a SQL Server instance-level setting. This setting should not be changed from the value of “1”. Changing this to anything other than “1” can have a significant negative impact on the BizTalk Server stored procedures and performance. If changing the parallelism setting for an instance of SQL Server will have an adverse effect on other database applications that are being executed on the SQL Server instance, you should create a separate instance of SQL Server dedicated to hosting the BizTalk Server databases.

Parallel queries are generally best suited to batch processing and decision support workloads. They are typically not desirable in a transaction processing environment where you have many short, fast queries running in parallel. In addition, changing the MDOP setting sometimes causes the query plan to be changed, which leads to poor query performance or even deadlocks with the BizTalk Server queries.

The BizTalk Server stored procedures provide the correct joins and lock hints wherever possible in order to try to keep the query optimizer from doing much work and changing the plan. These stored procedures provide consistent query executions by constructing the queries such that the query optimizer is taken out of the picture as much as possible.

For more information, see Microsoft Knowledge Base article 899000, ["The Parallelism setting for the instance of SQL Server when you configure BizTalk Server"](#) (<http://go.microsoft.com/fwlink/?LinkId=153432>).

SQL Server Statistics on the MessageBox Database

The following options are turned off by default in the BizTalk Server MessageBox database when it is created:

- Auto create statistics
- Auto update statistics

Do not enable these options on MessageBox databases. Enabling the "auto create statistics" and "auto update statistics" options can cause undesirable query execution delays, especially in a high-load environment.

In addition, the BizTalk Server stored procedures have exact joins and lock hints specified on the queries. This is done to ensure that the optimal query plan is used by the BizTalk Server queries in SQL Server. The distributions and expected results for the queries are known; the approximate number of rows returned is known. Statistics are generally not needed.

For more information, see the following Microsoft Knowledge Base articles:

- **912262**—"[The auto update statistics option, the auto create statistics option, and the Parallelism setting are turned off in the SQL Server database instance that hosts the BizTalk Server BizTalkMsgBoxDB database](http://go.microsoft.com/fwlink/?LinkId=153430)" (<http://go.microsoft.com/fwlink/?LinkId=153430>).
- **917845**—"[You experience blocking, deadlock conditions, or other SQL Server issues when you try to connect to the BizTalkMsgBoxDb database in BizTalk Server](http://go.microsoft.com/fwlink/?LinkId=153429)" (<http://go.microsoft.com/fwlink/?LinkId=153429>).

Changes to the MessageBox Database

The MessageBox database should be treated like non-Microsoft application source code. That is, you should not "tweak" the MessageBox database via changes to tables, indexes, stored procedures, and most SQL Server database settings. For more information, in the BizTalk Core Engine's WebLog, see the entry "[What you can and can't do with the MessageBox Database server](http://go.microsoft.com/fwlink/?LinkId=101577)" (<http://go.microsoft.com/fwlink/?LinkId=101577>).

Default Settings for the Database Index Rebuilds and Defragmentation

BizTalk Server does not support defragmenting indexes. "DBCC INDEXDEFRAG" and "ALTER INDEX ... REORGANIZE ..." are not supported since they use page locking, which can cause blocking and deadlocks with BizTalk Server. BizTalk Server does support database index rebuilds ("DBCC DBREINDEX" and "ALTER INDEX ... REBUILD ..."), but they should only be done during maintenance windows when BizTalk Server is not processing data. Index rebuilds while BizTalk Server is processing data are not supported.

For more information, see Microsoft Knowledge Base article 917845 "[You experience blocking, deadlock conditions, or other SQL Server issues when you try to connect to the BizTalkMsgBoxDb database in BizTalk Server](http://go.microsoft.com/fwlink/?LinkId=153429)" (<http://go.microsoft.com/fwlink/?LinkId=153429>).

Index fragmentation is not as much of a performance issue for BizTalk Server as it would be for a DSS system or an OLTP system that performs index scans. BizTalk Server does very selective queries and updates and BizTalk Server stored procedures should not cause table or index scans.

For more information about index fragmentation and workload types, see "[Microsoft SQL Server 2000 Index Defragmentation Best Practices](http://go.microsoft.com/fwlink/?LinkId=101580)" (<http://go.microsoft.com/fwlink/?LinkId=101580>). A quote from the article:

"As shown in Figure 1, there is little difference between the performance of the stored procedures before and after defragmenting. Because the underlying queries issued by these stored procedures acted upon very selective portions of the data, workload performance was not adversely affected by fragmented indexes."

**Note**

The contents of the article also apply to SQL Server 2008 SP1 and SQL Server 2005.

See Also

[Checklist: Configuring SQL Server](#)

Monitoring and Reducing Database I/O Contention

BizTalk Server performance is often predicated upon SQL Server performance, which in turn is often predicated upon disk I/O performance. Therefore, you should monitor and performance-tune disk I/O on the computers running SQL Server that house the BizTalk Server databases.

Monitoring Disk I/O

Because of the database-intensive nature of BizTalk Server, disk I/O can easily become a bottleneck on the MessageBox and BizTalk Tracking databases; this is true even if disk I/O has not previously been a bottleneck for the database files in your SQL Server environment. Thus, we recommend that you proactively measure disk I/O performance for the disks that house the data and transaction log files. For more information about monitoring disk I/O performance using System Monitor, see the SQL Server article "[Predeployment I/O Best Practices](#)" (<http://go.microsoft.com/fwlink/?LinkId=104829>). If you are using a SAN, you may also need specific tools from the SAN hardware manufacturer to measure disk I/O performance.

Separating the MessageBox and BizTalk Tracking (DTA) Databases and Log Files

Since the MessageBox and BizTalk Tracking databases are the most active, we recommend that you place the data files and transaction log files for each of these on dedicated drives to reduce the likelihood of problems with disk I/O contention. For example, you would need four drives for the MessageBox and BizTalk Tracking database files; one drive for each of the following:

- MessageBox data files
- MessageBox transaction log files
- BizTalk Tracking data files
- BizTalk Tracking transaction log files

Separating the MessageBox and BizTalk Tracking databases and separating the database files and transaction log files on different physical disks are considered best practices for reducing disk I/O contention. Try to spread the disk I/O across as many physical spindles as possible. For more information about avoiding disk contention, see [How to Avoid Disk Contention](#) (<http://go.microsoft.com/fwlink/?LinkId=158809>) in the BizTalk Server Performance Optimization Guide.

You should separate the files manually after configuring BizTalk Server. For more information, see the [BizTalk Server Database Optimization white paper](#) (<http://go.microsoft.com/fwlink/?LinkId=101578>).

See Also

[Using the Performance Analysis of Logs \(PAL\) Tool](#)

Reviewing and Testing SQL Server Cluster Configuration for Failover Scenarios

Windows Clustering and SQL Server allow you to run SQL Server in Active/Active mode where each node of the cluster is “active” and running one or more SQL Server instances. This would allow you, for example, to have the MessageBox database on one node and all other BizTalk Server databases on the other node. This allows you to maximize cluster hardware usage.

If you use this configuration, however, you must verify that each node can simultaneously handle the load of all SQL Server instances during a SQL Server cluster node failover.

Evaluating Failover for an Active/Active Cluster

Considerations when verifying that a single node can handle the load of all SQL Server instances in the event of a SQL Server cluster node failover include:

- Does the failover node have sufficient CPU resources?
- Does the failover node have sufficient memory?
- Is there sufficient network bandwidth?
- Can the failover node handle the increased disk I/O contention?

The following scenarios should be evaluated when testing failover:

- Power failure on the active server
- Power failure on the passive server
- Loss of disk connection
- Broken public network connection on the Active node
- Broken private network connection on the Active node
- Broken public network connection on the Passive node
- Broken private network connection on the Passive node
- Failed SQL Server service
- Failed SQL Server Agent service

Using an Active/Active/Passive Cluster

If you determine that one node cannot handle all SQL Server instances in a failover scenario, an alternative is to use an Active/Active/Passive clustering model. The Active/Active/Passive clustering model greatly increases the likelihood that there will always be one Passive node available for failover scenarios.

See Also

[Checklist: Configuring SQL Server](#)

Defining Auto-Growth Settings for Databases

You should set database auto-growth to a fixed number of megabytes instead of to a percentage, especially for the MessageBox and BizTalk Tracking databases. Depending on your BizTalk application and throughput, the MessageBox and Tracking databases can get quite large. If you set auto-growth to a percentage, then the auto-growth can be substantial as well.

How Instant File Initialization Works

When SQL Server increases the size of a file, it must first initialize the new space before it can be used. This is a blocking operation that involves filling the new space with empty pages. SQL Server 2005 or later running on Windows Server 2003 or later supports “instant file initialization.” This can greatly reduce the performance impact of a file growth operation. For more information, see ["Database File Initialization"](http://go.microsoft.com/fwlink/?LinkId=101579) (http://go.microsoft.com/fwlink/?LinkId=101579) in the SQL Server documentation. This topic outlines the steps that must be taken to enable instant file initialization.

Enabling Instant File Initialization

For steps on enabling instant file initialization, see ["Database File Initialization"](http://go.microsoft.com/fwlink/?LinkId=101579) (http://go.microsoft.com/fwlink/?LinkId=101579) in the SQL Server documentation. For creating file groups and moving the BizTalk Server database tables, indexes, and log files into the appropriate file groups, follow the recommendations in "Appendix B - Recommended BizTalk Server Database Configuration" in the ["BizTalk Server Database Optimization" white paper](http://go.microsoft.com/fwlink/?LinkId=101578) (http://go.microsoft.com/fwlink/?LinkId=101578). Ideally the size of files supporting the file groups should be pre-allocated and if possible, set to a static size.

If the system is new and the static sizes have not been definitively established, then configure files with the **Enable Autogrowth** option and specify file growth **In Megabytes**. The growth increment should generally be no larger than 100 MB (for large files), 10 MB (for medium-sized files), or 1 MB (for small files).

Backing Up Databases

Because BizTalk Server uses distributed transactions across multiple databases, the Backup BizTalk Server job creates synchronized backups of all BizTalk Server databases. This is accomplished by using marked transactions with the “Full” database recovery model. This is required for the backups to be transactionally consistent across databases. For more information, see ["Marked Transactions, Full Backups, and Log Backups"](http://go.microsoft.com/fwlink/?LinkId=151565) (http://go.microsoft.com/fwlink/?LinkId=151565) in the BizTalk Server documentation.

Advantages of the Backup BizTalk Server Job

The Backup BizTalk Server job is the only supported method for backing up the BizTalk Server databases. Use of SQL Server jobs to back up the BizTalk Server databases in a production environment is not supported.

If the Backup BizTalk Server job is not run, the BizTalk Server database transaction logs will grow unbounded. The backup job truncates the transaction logs, which keep them from growing unbounded. If the BizTalk Server database transaction logs continue to grow, they could at some point fill the disk they are housed on.



Note

Using both the Backup BizTalk Server job and log shipping is currently the only fully documented and supported method for performing BizTalk Server database backup and restore.

Guidelines for the Backup BizTalk Server Job

You should restore the entire BizTalk Server environment on a regular basis. This includes not only the SQL server(s) and the BizTalk Server databases, but also the computers running BizTalk Server. You should document and practice these procedures before a failure actually occurs.



Note

The Backup BizTalk Server job does not delete old backup files. You must define and put processes in place to archive the old backup files and move them from the backup directory that the backup job uses.

Additional Resources

See the following topics in the BizTalk Server documentation:

- For more information about specific BizTalk Server backup and restore tasks, see "[Checklist: Backup and Restore](http://go.microsoft.com/fwlink/?LinkId=154070)" (<http://go.microsoft.com/fwlink/?LinkId=154070>).
- For an overview of Backing up and restoring BizTalk Server, see "[Backing Up and Restoring BizTalk Server](http://go.microsoft.com/fwlink/?LinkId=154071)" (<http://go.microsoft.com/fwlink/?LinkId=154071>).
- For more information about configuring the Backup BizTalk Server job, see "[How to Configure the Backup BizTalk Server Job](http://go.microsoft.com/fwlink/?LinkId=154072)" (<http://go.microsoft.com/fwlink/?LinkId=154072>).

See Also

[Using BizTalk Server Log Shipping for Disaster Recovery](#)

Using BizTalk Server Log Shipping for Disaster Recovery

BizTalk Server implements database standby capabilities through the use of database log shipping. BizTalk Server log shipping automates the backup and restore of database and transaction log files, allowing a standby server to resume database processing in the event that the production database server fails.

How Log Shipping Works

The SQL Server Agent jobs used by BizTalk Server for log shipping synchronize data between the source server used in production and the destination server used as a standby every 15 minutes by default (every time that the Backup BizTalk Server job runs).

Using Log Shipping for Disaster Recovery

Do the following when using BizTalk Server log shipping for disaster recovery:

- Follow the steps in the topic [Checklist: Increasing Availability with Disaster Recovery](#) to increase availability of a production BizTalk Server environment using disaster recovery.
- Verify that the disaster recovery servers have the capacity to handle production load. Ensure that the standby servers have the same or similar resources available (CPU/memory/disk) as the production servers.
- Ensure that the specifics of your disaster recovery routine are well documented. Document every step of your disaster recovery preparation and implementation in detail. Disaster seldom strikes when it is convenient so assume that the parties responsible for implementing the disaster recovery procedure are starting their first day of work and will be doing this for the first time.

- As part of regular testing, practice failover to the disaster recovery site, especially as new BizTalk applications are put in production.

Perform failover testing as a part of regular testing and maintenance to ensure that it can be performed smoothly.

See Also

[Disaster Recovery](#)

Monitoring SQL Server Agent Jobs

BizTalk Server includes multiple SQL Server Agent jobs that perform important functions to keep your servers operational and healthy. You should monitor the health of these jobs and ensure that they are running without errors.

Guidelines for Monitoring the SQL Server Agent Jobs

Following are guidelines for monitoring the jobs:

- **Verify that the SQL Server Agent service is running**
- **Verify that the SQL Server Agent jobs installed by BizTalk Server are enabled and running successfully**

The BizTalk Server SQL Server Agent jobs are crucial: if they are not running, system performance will degrade over time.

- **Verify that the BizTalk Server SQL Server Agent jobs are completing in a timely manner**

Set up Microsoft System Center Operations Manager to monitor the jobs.

You should be aware of schedules that are particular to certain jobs:

- The `MessageBox_Message_ManageRefCountLog_BizTalkMsgBoxDb` job runs continuously by default. Monitoring software should take this schedule into account and not produce warnings.
- The `MessageBox_Message_Cleanup_BizTalkMsgBoxDb` job is not enabled or scheduled, but it is started by the `MessageBox_Message_ManageRefCountLog_BizTalkMsgBoxDb` job every 10 seconds. Therefore, this job should not be enabled, scheduled, or manually started.
- **Verify that the Startup type of the SQL Server Agent service is configured correctly**
Verify that the SQL Server Agent service is configured with a **Startup type of Automatic** unless the SQL Server Agent service is configured as a cluster resource on a Windows Server cluster. If the SQL Server Agent service is configured as a cluster resource, then you should configure the **Startup type** as **Manual** since the service will be managed by the Cluster service.

Additional Resources

- For more information about monitoring the BizTalk Server SQL Agent jobs, see [Monitoring SQL Server Agent Jobs and Databases](#).
- For more information about the SQL Server Agent jobs that are included with BizTalk Server see ["Database Structure and Jobs"](http://go.microsoft.com/fwlink/?LinkID=153451) (<http://go.microsoft.com/fwlink/?LinkID=153451>).

Purging and Archiving Tracking Data

It is important to configure and enable the DTA Purge and Archive SQL Agent job. This job archives and purges old data from the BizTalk Tracking (DTA) database. This is essential for a healthy BizTalk Server system. A large Tracking database will begin to affect the performance of the tracking host and any other processes that query the Tracking database. If the tracking data is not purged from the Tracking database, the database will continue to grow.

Guidelines for Using the DTA Purge and Archive Job

- **Ensure that the DTA Purge and Archive SQL Agent job is properly configured, enabled, and successfully completing.**

This job is not enabled by default because you must first configure it to include a directory where the archive files can be written.

- **If you only need to purge the old data and do not need to archive it first, then change the SQL Agent job to call the stored procedure “dtasp_PurgeTrackingDatabase”.**

This skips the archive step, and just does the purge. For more information about this stored procedure and changing the SQL Agent job to use it, see ["How to Purge Data from the BizTalk Tracking Database"](http://go.microsoft.com/fwlink/?LinkId=153817) (http://go.microsoft.com/fwlink/?LinkId=153817).

- **Ensure that the job is able to purge the tracking data as fast as the incoming tracking data is generated.**

It is acceptable for the job to get behind during peak load times, but it should always be able to catch up. If the purge job gets behind and is never able to catch up, the Tracking database will continue to grow, and performance will eventually be adversely affected.

- **Review the soft purge and hard purge parameters to ensure that you are keeping data for the optimal length of time.**

For more information about these parameters see ["Archiving and Purging the BizTalk Tracking Database"](http://go.microsoft.com/fwlink/?LinkId=153816) (http://go.microsoft.com/fwlink/?LinkId=153816).

- **If you need to keep the tracking archive files, ensure that you have a process in place to successfully restore and use them.**

See Also

[Checklist: Configuring SQL Server](#)

Checklist: Configuring BizTalk Server

Follow these steps when preparing BizTalk Server for use in a BizTalk Server production environment.

Steps	Reference
Configure BizTalk hosts and host instances.	Separate sending, receiving, processing, and tracking functionality into multiple hosts. This provides flexibility when configuring the workload and enables you to stop one host without affecting other hosts. For more

Steps	Reference
	information, see Configuring Hosts and Host Instances .
Make sure you adhere to and understand the maximum practical limits of memory usage for a BizTalk host instance.	Maximum Practical Limits of Memory Usage of a 32-bit BizTalk Host Instance
Configure a dedicated tracking host.	Use a dedicated host that does nothing but host tracking. This prevents hosting tracking from having an impact on the performance of other BizTalk artifacts running in the same host. It also allows you to stop other hosts without interfering with tracking. The tracking host should be run on at least two computers running BizTalk Server (for redundancy in case one fails). For more information, see Configuring a Dedicated Tracking Host .
Setting SOAP, HTTP, and HTTP-based WCF Adapters Concurrent Connections	Apply IIS Configuration Settings
Implement a BizTalk application upgrade and versioning strategy.	<ul style="list-style-type: none"> • If you need to support long-running orchestrations, and/or you need to perform BizTalk application deployments with no BizTalk application downtime, then you need to implement and practice a solid, end-to-end BizTalk Server versioning strategy for the different versioning scenarios. • If you need to support long-running orchestrations, side-by-side deployments, or no-downtime upgrades, then you should implement an assembly versioning and packaging strategy that includes factoring. <p>For more information, see Upgrading and Versioning Strategies for Applications.</p>
Script BizTalk Server application deployments.	<p>BizTalk application deployments should be scripted where possible. You should document with detailed steps anything that you do not script. For more information, see:</p> <ul style="list-style-type: none"> • Using Scripts to Deploy Applications • Managing Applications
Predefine processes for resubmitting messages	Establish and document a procedure to check

Steps	Reference
and restarting workflows.	for suspended service instances and take appropriate actions. In most BizTalk Server environments, this should be performed as a part of the daily maintenance of your BizTalk Server environment. For more information about performing daily maintenance checks, see Checklist: Performing Daily Maintenance Checks .
Define escalation paths for issues that may be encountered in a BizTalk Server environment.	<ul style="list-style-type: none"> • Determine roles and responsibilities • Define escalation process and paths • Define “short-circuit” processes and paths when necessary for “critical situation” scenarios • Define an escalation path for vendor issues, including Microsoft, other software vendors, hardware vendors (for example, servers, SAN, switches)
Adhere to certain considerations when using BizTalk Server on a 64-bit Windows operating system	Considerations While Using BizTalk Server on a 64-bit Windows Operating System
Adhere to the best practices for BizTalk Server settings.	Best Practices for BizTalk Server Settings

In This Section

- [Configuring Hosts and Host Instances](#)
- [Configuring a Dedicated Tracking Host](#)
- [Apply IIS Configuration Settings](#)
- [Upgrading and Versioning Strategies for Applications](#)
- [Using Scripts to Deploy Applications](#)
- [Considerations While Using BizTalk Server on a 64-bit Windows Operating System](#)
- [Best Practices for BizTalk Server Settings](#)

Configuring Hosts and Host Instances

A BizTalk Host represents a logical set of zero or more run-time processes in which you can deploy BizTalk Server services and artifacts (such as adapter handlers, receive locations, and orchestrations). A host instance is the physical instance of a host on a computer running BizTalk Server. For more information about BizTalk Hosts and host instances, see [Hosts](#) (<http://go.microsoft.com/fwlink/?LinkId=154189>) and [Host Instances](#) (<http://go.microsoft.com/fwlink/?LinkId=154190>).

For more information about managing BizTalk Hosts and host instances, see [Managing BizTalk Hosts and Host Instances](http://go.microsoft.com/fwlink/?LinkId=154191) (http://go.microsoft.com/fwlink/?LinkId=154191).

For information about how to configure a dedicated tracking host, see [Configuring a Dedicated Tracking Host](#).

Separating Host Instances by Functionality

In addition to the high availability aspects of the host instance configuration, you should separate sending, receiving, processing, and tracking functionality into multiple hosts. This provides flexibility when configuring the workload in your BizTalk group and is the primary means of distributing processing across a BizTalk group. This also allows you to stop one host without affecting other hosts. For example, you may want to stop sending messages to let them queue up in the MessageBox database, while still allowing the inbound receiving of messages to occur.

Separating host instances by functionality also provides the following benefits:

- Each host instance has its own set of resources such as memory, handles, and threads in the .NET thread pool.
- Multiple BizTalk Hosts will also reduce contention on the MessageBox database host queue tables since each host is assigned its own work queue tables in the MessageBox database.
- Throttling is implemented in BizTalk Server at the host level. This allows you to set different throttling characteristics for each host.
- Security is implemented at the host level; each host runs under a discrete Windows identity. This would allow you, for example, to give **Host_A** access to **FileShare_B**, while not allowing any of the other hosts to access the file share.



Note

While there are benefits to creating additional host instances, there are also potential drawbacks if too many host instances are created. Each host instance is a Windows service (BTSNTSvc.exe or BTSNTSvc64.exe), which generates additional load against the MessageBox database and consumes computer resources (such as CPU, memory, threads).

For more information about modifying BizTalk Server Host properties, see [How to Modify Host Properties](http://go.microsoft.com/fwlink/?LinkId=154192) (http://go.microsoft.com/fwlink/?LinkId=154192).

Maximum Practical Limits of Memory Usage of a 32-bit BizTalk Host Instance

32-bit processes on 32-bit Windows operating system with /3GB set have 3 gigabytes (GB) of addressable memory if the process is "large address aware" (that is, the executable has the IMAGE_FILE_LARGE_ADDRESS_AWARE flag set in the image header). The BizTalk host process, being "large address aware", can address 3 GB of memory on Windows operating system with /3GB set. Similarly, 32-bit processes on 64-bit Windows operating system (AMD64) have 4 GB of addressable memory, if the process is "large address aware". Again, the BizTalk host process, being "large address aware", can address 4 GB of memory when running as a 32-bit process on 64-bit Windows operating system. 64-bit processes on 64-bit Windows operating system (AMD64) have 8 terabytes of addressable memory.

Even though the maximum memory addressable by a process on a 32-bit Windows operating system (without the /3GB switch) is 2 GB, a .NET application (such as a BizTalk host instance)

will receive out of memory errors before the "virtual bytes" reaches 2 GB. The table below summarizes this and includes the practical limits for virtual bytes and private bytes.

Process	Windows OS	Addressable Memory (with a Large Address Aware process)	Practical Limit for Virtual Bytes	Practical Limit for PrivateBytes
32-bit	32-bit	2 GB	1400 MB	800 MB
32-bit	32-bit with /3GB	3 GB	2400 MB	1800 MB
32-bit	64-bit	4 GB	3400 MB	2800 MB
64-bit	64-bit	8 terabytes	-	-

For more information, see:

- [ASP.NET Performance Monitoring, and When to Alert Administrators](http://go.microsoft.com/fwlink/?LinkId=151856) (http://go.microsoft.com/fwlink/?LinkId=151856)
- [Memory Limits for Windows Releases](http://go.microsoft.com/fwlink/?LinkId=151857) (http://go.microsoft.com/fwlink/?LinkId=151857)

See Also

[Checklist: Configuring BizTalk Server](#)
[Configuring a Dedicated Tracking Host](#)

Configuring a Dedicated Tracking Host

BizTalk Server is optimized for throughput, so the main orchestration and messaging engines do not actually move events or messages directly to the BizTalk Tracking (DTA) or Business Activity Monitoring (BAM) databases, since this would divert these engines from their primary job of executing business processes. Instead, BizTalk Server leaves the events and messages in the MessageBox database and marks them as requiring a move to the BizTalk Tracking or BAM databases. A background process (the tracking host) then moves the events to the BizTalk Tracking and BAM databases, while a SQL Server Agent job copies tracked messages to the BizTalk Tracking database.

Advantages of Using a Dedicated Tracking Host

A BizTalk Host that hosts tracking is responsible for moving the DTA and BAM tracking data from the MessageBox database to the BizTalk Tracking (DTA) and BAM Primary Import databases. This movement of tracking data has an impact on the performance of other BizTalk artifacts that are running in the same host that is hosting tracking. Thus, you should use a dedicated host that does nothing but host tracking.

Using a dedicated tracking host also allows you to stop other BizTalk hosts without interfering with BizTalk Server tracking. The movement of tracking data out of the MessageBox database is critical for a healthy BizTalk Server system. If the BizTalk Host responsible for moving tracking

data in the BizTalk group is stopped, the Tracking Data Decode service will not run. The impact of this is as follows:

- HAT tracking data will not be moved from the MessageBox database to the BizTalk Tracking database.
- BAM tracking data will not be moved from the MessageBox database to the BAM Primary Import database.
- Because data is not moved, it cannot be deleted from the MessageBox database.
- When the Tracking Data Decode service is stopped, tracking interceptors will still run and write tracking data to the MessageBox database. If the data is not moved, this will cause the MessageBox database to become bloated, which will affect performance over time. Even if custom properties are not tracked or BAM profiles are not set up, by default some data is tracked (such as pipeline receive / send events and orchestration events). If you do not want to run the Tracking Data Decode service, turn off all tracking so that no interceptors save data to the database. To disable global tracking, see [How to Turn Off Global Tracking](http://go.microsoft.com/fwlink/?LinkId=154193) (<http://go.microsoft.com/fwlink/?LinkId=154193>) Use the BizTalk Server Administration console to selectively disable tracking events.

Optimizing Performance for a Dedicated Tracking Host

This host should be run on at least two computers running BizTalk Server (for redundancy in case one fails). For optimal performance, you should have at least one tracking host instance per MessageBox database. The actual number of tracking host instances should be $N + 1$, where N = the number of MessageBox databases. The "+ 1" is for redundancy. There is no benefit to adding more than that, because only one tracking host instance can move data for a specific MessageBox database. As a result, locking should never be an issue. The one additional tracking host instance is added for fault tolerance; if one of the tracking host instances fails, the additional instance will assume the duties of the failed instance.

A tracking host instance moves tracking data for specific MessageBox databases, but there will never be more than one tracking host instance moving data for a specific MessageBox database. For example, if you have three MessageBox databases, and only two tracking host instances, then one of the host instances needs to move data for two of the MessageBox databases. Adding a third tracking host instance distributes the tracking host work to another computer running BizTalk Server. In this scenario, adding a fourth tracking host instance would not distribute any more tracking host work, but would provide an extra tracking host instance for fault tolerance.

For more information about the BAM Event Bus service, see the following topics in BizTalk Server 2010 Help:

- [Managing the BAM Event Bus Service](http://go.microsoft.com/fwlink/?LinkId=154194) (<http://go.microsoft.com/fwlink/?LinkId=154194>)
- [Creating Instances of the BAM Event Bus Service](http://go.microsoft.com/fwlink/?LinkId=154195) (<http://go.microsoft.com/fwlink/?LinkId=154195>)

Configuring a Dedicated Tracking Host

To perform the procedure in this section, you must have the following user rights for modifying host properties to allow host tracking:

- You must be a member of the BizTalk Server Administrators group.
- You must have the following rights in SQL Server:

- You must be either a SQL Server administrator or a member of the *db_owner* or *db_securityadmin* SQL Server database roles in the BizTalk Tracking database (BizTalk DTADb), MessageBox databases (BizTalkMsgBoxDb), and the BAM Primary Import database (BAMPrimaryImport).
- You must be a member of the sysadmin SQL Server role on all the computers where there are MessageBox databases, or a member of the *db_owner* or *db_ddladmin* SQL Server role for all the MessageBox databases.

► **To enable host tracking**

1. Click **Start**, click **Programs**, click **Microsoft BizTalk Server 2010**, and then click **BizTalk Server Administration**.
2. In the console tree, expand **BizTalk Server 2010 Administration**, expand the BizTalk group, click **Platform Settings**, and then click **Hosts**.
3. In the details pane, right-click the host that you want to modify, and then click **Properties**.
4. In the **Host Properties** dialog box, on the **General** tab, select or clear **Options - Allow Host Tracking**, and then click **OK**.

Select this check box to indicate that the host loads the BizTalk Tracking component to process health monitoring and business data. If you select this check box, the current host will have read/write access to the tracking tables in the MessageBox database, as well as to the Tracking database. Accordingly, any objects running in this host will also have read/write access to these databases.

If you clear the check box, the host will have only write access to the tracking tables in the MessageBox database and will not have access to the Tracking database.

See Also

[Checklist: Configuring BizTalk Server](#)

Apply IIS Configuration Settings

By default the SOAP, HTTP, and HTTP-based WCF adapters (and .NET in general) open only two concurrent HTTP connections from each BizTalk host instance to any specific destination server. For example, if you have a SOAP send port sending messages to <http://www.contoso.com/SomeWebService.asmx>, then by default each host instance running on each BizTalk Server will open only two concurrent HTTP connections to www.contoso.com, no matter how many messages need to be sent.

This setting conforms to the IETF RFC for the HTTP 1.1 specification, and although it is suitable for user scenarios, it is not optimized for high throughput server to server communications. The default setting effectively throttles outbound SOAP and HTTP calls to each destination server to two concurrent sends from each BizTalk Server host instance.

Configure ASP.NET MaxConcurrentRequests for IIS 7.0 Integrated mode

When ASP.NET 2.0 is hosted on IIS 7.0 in Integrated Mode, the use of threads is handled differently than on IIS 6.0 or on IIS 7.0 in Classic Mode. When ASP.NET 2.0 is hosted on IIS 7.0 in Integrated mode, ASP.NET 2.0 restricts the number of concurrently executing requests instead

of the number of threads concurrently executing requests. For synchronous scenarios, this will indirectly limit the number of threads because the number of requests will be the same as the number of threads. But for asynchronous scenarios, the number of requests and threads will likely be very different because you could have far more requests than threads. When you run ASP.NET 2.0 on IIS 7.0 in integrated mode, the `minFreeThreads` and `minLocalRequestFreeThreads` of the “httpRuntime” element in the `machine.config` are ignored. For IIS 7.0 Integrated mode, a DWORD named `MaxConcurrentRequestsPerCPU` within `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\ASP.NET\2.0.50727.0` determines the number of concurrent requests per CPU. By default, the registry key does not exist and the number of requests per CPU is limited to 12. .NET Framework 3.5 SP1 includes an update to the v2.0 binaries that supports configuring IIS application pools via the `aspnet.config` file. This configuration applies to integrated mode only (Classic/ISAPI mode ignores these settings). The new `aspnet.config` config section with default values is listed below:

```
<system.web>
  <applicationPool maxConcurrentRequestsPerCPU="12" maxConcurrentThreadsPerCPU="0"
requestQueueLimit="5000"/>
</system.web>
```

As a rule of thumb, `MaxConcurrentRequestsPerCPU` should be set to a very large value, such as 5000.

In IIS 7.0 Integrated Mode, the `maxWorkerThreads` and the `maxIoThreads` parameters in the “processModel” section of the `machine.config` file are not used to govern the number of running requests, per se, but they are still used to govern the size of the CLR thread pool used by ASP.NET. When the “processModel” section of the `machine.config` has “`autoConfig=true`” (which is the default setting), this will give the application pool up to 100 worker threads (`MaxWorkerThreads`) per logical CPU. So a common commodity server with 2 dual-core CPUs would have 400 `MaxWorkerThreads`. This should be sufficient for all but the most demanding applications.

For more information about configuring ASP.NET Thread Usage on IIS 7.0 and 6.0, see [Thomas Marquardt's Blog on ASP.NET Thread Usage on IIS 7.0 and 6.0](http://go.microsoft.com/fwlink/?LinkId=157518) (<http://go.microsoft.com/fwlink/?LinkId=157518>).

Log only essential information or completely disable IIS logging

IIS logging should be minimized or even disabled in a production environment. To disable logging follow these steps:

1. Click **Start**, point to **All Programs**, click **Administrative Tools**, and then click **Internet Information Services (IIS) Manager**.
2. In the **Connections** pane, click to expand **Sites**, click to select the Web site for which you would like to disable logging, click to select **Features View**, and then double-click the **Logging** feature.
3. Click **Disable** in the **Actions** pane to disable logging for this Web site.

Disable IIS ASP debugging in production environments

IIS ASP debugging should be disabled in a production environment. To disable IIS ASP debugging follow these steps:

1. Click **Start**, point to **All Programs**, click **Administrative Tools**, and then click **Internet Information Services (IIS) Manager**.
2. In the **Connections** pane, click to expand **Sites**, click to select the web site for which you would like to disable ASP debugging, click to select **Features View**, and then double-click the **ASP** feature.
3. Click to expand **Compilation**, click to expand **Debugging Properties**, and verify that both **Enable Client-side Debugging** and **Enable Server-side Debugging** are set to **False**.
4. If necessary, click **Apply** in the **Actions** pane.

Disable debugging for ASP.NET Applications and Web Services by specifying the <compilation debug="false"/> section in the web.config file for the web application.

Tune the value of the ASP Threads Per Processor Limit property

The **ASP Threads Per Processor Limit** property specifies the maximum number of worker threads per processor that IIS creates. Increase the value for the Threads Per Processor Limit until the processor utilization meets at least 50 percent or above. This setting can dramatically influence the scalability of your Web applications and the performance of your server in general. Because this property defines the maximum number of ASP requests that can execute simultaneously, this setting should remain at the default value unless your ASP applications are making extended calls to external components. In this case, you may increase the value of Threads Per Processor Limit. Doing so allows the server to create more threads to handle more concurrent requests. The default value of Threads Per Processor Limit is 25. The maximum recommended value for this property is 100.

To increase the value for the Threads Per Processor Limit follow these steps:

1. Click **Start**, point to **All Programs**, click **Administrative Tools**, and then click **Internet Information Services (IIS) Manager**.
2. In the **Connections** pane, select the web server, click to select **Features View**, and then double-click the **ASP** feature.
3. Click to expand **Limits Properties** under **Behavior**, click **Threads Per Processor Limit**, enter the desired value for **Threads Per Processor Limit** and click **Apply** in the **Actions** pane.

For more information about how to modify the properties in the <limits> element of the IIS 7.0 <asp> element, see [ASP Limits <limits>](http://go.microsoft.com/fwlink/?LinkId=157483) (<http://go.microsoft.com/fwlink/?LinkId=157483>).



Note

Because this property can only be applied at the server level, modification of this property affects all Web sites that run on the server.



Note

The **ASP Threads Per Processor Limit** property for IIS 7.0 replaces the IIS 6.0 **ASPPProcessorThreadMax** ASP Metabase setting. For information about the IIS 6.0

ASPProcessorThreadMax ASP Metabase setting, see [Tuning ASP Metabase Settings](http://go.microsoft.com/fwlink/?LinkId=158834) (http://go.microsoft.com/fwlink/?LinkId=158834) on MSDN.

Tune the value of the ASP Queue Length property

The goal of tuning this property is to ensure good response time while minimizing how often the server sends the HTTP 503 (Server Too Busy) error to clients when the ASP request queue is full. If the value of ASP Queue Length property is too low, the server will send the HTTP 503 error with greater frequency. If the value of ASP Queue Length property is too high, users might perceive that the server is not responding when in fact their request is waiting in the queue. By watching the queue during periods of high traffic, you should discern a pattern of web request peaks and valleys. Make note of the peak value, and set the value of the ASP Queue Length property just above the peak value. Use the queue to handle short-term spikes, ensure response time, and throttle the system to avoid overload when sustained, unexpected spikes occur. If you do not have data for adjusting the ASP Queue Length property, a good starting point will be to set a one-to-one ratio of queues to total threads. For example, if the ASP Threads Per Processor Limit property is set to 25 and you have four processors ($4 * 25 = 100$ threads), set the ASP Queue Length property to 100 and tune from there.

To increase the value for the Queue Length property, follow these steps:

1. Click **Start**, point to **All Programs**, click **Administrative Tools**, and then click **Internet Information Services (IIS) Manager**.
2. In the **Connections** pane, select the Web server, click to select **Features View**, and then double-click the **ASP** feature.
3. Click to expand **Limits Properties** under **Behavior**, click **Queue Length**, enter the desired value for **Queue Length** and then click **Apply** in the **Actions** pane.

For more information about how to modify the properties in the <limits> element of the IIS 7.0 <asp> element, see [ASP Limits <limits>](http://go.microsoft.com/fwlink/?LinkId=157483) (http://go.microsoft.com/fwlink/?LinkId=157483).



Note

Because this property can only be applied at the server level, modification of this property affects all Web sites that run on the server.



Note

The ASP **Queue Length** property for IIS 7.0 replaces the IIS 6.0 **AspRequestQueueMax** ASP Metabase setting. For information about the IIS 6.0 **AspRequestQueueMax** ASP Metabase setting, see [Tuning ASP Metabase Settings](http://go.microsoft.com/fwlink/?LinkId=158834) (http://go.microsoft.com/fwlink/?LinkId=158834) on MSDN.

Tune the MaxPoolThreads registry entry

This setting specifies the number of pool threads to create per processor. Pool threads watch the network for requests and process incoming requests. The MaxPoolThreads count does not include threads that are consumed by ISAPI applications. Generally, you should not create more than 20 threads per processor. MaxPoolThreads is a REG_DWORD registry entry located at HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\InetInfo\Parameters\ with a default value of 4.

Disable WCF services tracing

Use the Configuration Editor Tool (SvcConfigEditor.exe) to disable WCF services tracing in a production environment. For more information about the Configuration Editor Tool, see [Configuration Editor Tool \(SvcConfigEditor.exe\)](http://go.microsoft.com/fwlink/?LinkID=127070) (http://go.microsoft.com/fwlink/?LinkID=127070).

See Also

[Checklist: Configuring BizTalk Server](#)

Upgrading and Versioning Strategies for Applications

BizTalk application versioning can become an issue when you need to run two versions of a BizTalk solution side-by-side, or if you cannot use BizTalk application downtime to deploy a new version. If you do not need to run two versions of the solution simultaneously (for example, where you have no long-running orchestrations), and service maintenance windows are available, then it is perfectly acceptable to undeploy the old version, and deploy the new version as a versioning strategy (no assembly versioning). This is a possible versioning strategy, although we still recommend incrementing the file version number (to let you know what version is deployed on the computers running BizTalk Server).

When to Use Versioning

If you need to support long-running orchestrations, and/or you need to perform BizTalk application deployments with no BizTalk application downtime, then you need to implement and practice a solid, end-to-end BizTalk Server versioning strategy for the different versioning scenarios. This includes .NET assembly versioning and versioning of all BizTalk artifacts, which includes schemas, maps, pipelines, pipeline components, orchestrations, custom adapters, custom classes called in orchestrations and maps, business rules, and BAM.

Schema versioning is unique in that the BizTalk Server pipelines determine the message type of a message based on the target namespace plus root node name defined in the schema. For more information, see [Schema Resolution in Pipeline Components](http://go.microsoft.com/fwlink/?LinkId=154207) (http://go.microsoft.com/fwlink/?LinkId=154207). If you need to version your schemas, a version indicator must be part of the target namespace. Changing the schema version has a ripple effect throughout your solution, and therefore should be planned in advance. When creating orchestration messages, follow the recommendations in the MSDN Magazine article [BizTalk Server: 8 Tips And Tricks For Better BizTalk Programming](http://go.microsoft.com/fwlink/?LinkId=101594), tip #1: "Always Use Multi-Part Message Types" (http://go.microsoft.com/fwlink/?LinkId=101594). Use of this method provides greater flexibility when versioning schemas.

Using Factoring for Assembly Versioning

If you need to support long-running orchestrations, side-by-side deployments, or no-downtime upgrades, then you should implement an assembly versioning and packaging strategy. In order to perform assembly versioning of BizTalk artifacts, your BizTalk solution assemblies need to be factored (packaged) in such a way to allow for BizTalk Server versioning. There are three types of factoring:

- **No factoring**

All BizTalk artifacts are in one assembly. This is the easiest to understand and deploy, but provides the least amount of flexibility.

- **Full factoring**

Each BizTalk artifact is in its own assembly. This provides the most flexibility, but is the most complex to deploy and understand.

- **Optimal factoring**

Somewhere in-between “no factoring” and “full factoring” based on in-depth analysis of your BizTalk applications. In addition to versioning, this allows you to easily implement your BizTalk Host design. This is achieved by looking for relationships among BizTalk artifacts. Artifacts that are always versioned together can typically be put in the same assembly. If independent versioning of the artifacts is required, then they must be put in different assemblies. This is the level of factoring you want to achieve.

Additional Resources

For more information about implementation and tuning best practices see [MSDN Webcast: Implementation and Tuning Best Practices for BizTalk Server Solutions](http://go.microsoft.com/fwlink/?LinkId=101595) (<http://go.microsoft.com/fwlink/?LinkId=101595>).

Define and practice a solid versioning strategy to ensure it provides any side-by-side deployment strategies you might need. Resources for BizTalk Server application upgrade and versioning strategies include the following:

- [Updating an Application](#)
- [Updating BizTalk Applications](http://go.microsoft.com/fwlink/?LinkId=154208) (<http://go.microsoft.com/fwlink/?LinkId=154208>) and subtopics.
- [BizTalk Server Project Versioning](http://go.microsoft.com/fwlink/?LinkId=154209) (<http://go.microsoft.com/fwlink/?LinkId=154209>)
- [Understanding BizTalk Server Application Deployment](http://go.microsoft.com/fwlink/?LinkId=101599) (<http://go.microsoft.com/fwlink/?LinkId=101599>)
- [BizTalk Server 2006 - Pipeline Component Side-By-Side deployment](http://go.microsoft.com/fwlink/?LinkId=101600) (<http://go.microsoft.com/fwlink/?LinkId=101600>)
- [BizTalk Server 2006 Lifecycle short video demos with description](http://go.microsoft.com/fwlink/?LinkId=101601) (<http://go.microsoft.com/fwlink/?LinkId=101601>)



Note

Even though some of these articles were written specifically for BizTalk Server 2006 R2, their content also applies to BizTalk Server 2010.

See Also

[Checklist: Configuring BizTalk Server](#)

Using Scripts to Deploy Applications

You should use scripts to deploy BizTalk applications where possible. Scripting reduces the risk of human error during the deployment process. You should also document your deployment procedures in depth. You should document anything that you do not script with very detailed steps. This includes documenting any changes to external systems and to deployment of non-Microsoft components.

Using BTSTask

You can use BTSTask.exe to script the creation of BizTalk applications, as well as to import existing BizTalk Server .msi packages. If the creation of the applications is scripted, then the packages can be automatically built using an automated process on a build server.

See the following topics for more information about scripting BizTalk application deployments:

- [Deploying and Managing BizTalk Applications](http://go.microsoft.com/fwlink/?LinkId=154210)
(<http://go.microsoft.com/fwlink/?LinkId=154210>)
- [Understanding BizTalk Server Application Deployment](http://go.microsoft.com/fwlink/?LinkId=101599)
(<http://go.microsoft.com/fwlink/?LinkId=101599>)



This latter article also applies to BizTalk Server 2010.

See Also

[Checklist: Configuring BizTalk Server](#)

Considerations While Using BizTalk Server on a 64-bit Windows Operating System

When using BizTalk Server on a 64-bit Windows operating system, make sure you consider the issues described in this topic. For frequently asked questions related to 64-bit support for Microsoft BizTalk Server 2010, see [BizTalk Server 64-bit Support](http://go.microsoft.com/fwlink/?LinkId=155306) (<http://go.microsoft.com/fwlink/?LinkId=155306>).

Modify the Process Memory Usage Throttling Threshold

By default, the **Process memory usage** host throttling threshold is set to 25. If this value is exceeded and the BizTalk process memory usage is more than 300 MB, a throttling condition may occur. On a 64-bit server, you can increase this value to 100. This allows for more memory consumption by the BizTalk process before throttling occurs. For instructions on how to modify the process memory usage host throttling threshold, see [How to Modify the Default Host Throttling Settings](http://go.microsoft.com/fwlink/?LinkId=157210) (<http://go.microsoft.com/fwlink/?LinkId=157210>).



The maximum available process memory is capped at an address space size of 2 GB for purposes of this calculation, even if the system has more than 2 GB of physical memory. A 2 GB process memory upper limit is used for purposes of this calculation on both 32 bit and 64 bit systems. In order to prevent out of memory errors, it is recommended that you do not specify a value that will allow host instance memory to exceed 1.54 GB when running a 32 bit version of BizTalk Server, regardless of if BizTalk Server is installed on a 32 bit or 64 bit operating system. For example, if you specify a value from 1 through 100 for this setting to initiate throttling based upon the percentage of process memory used, do not enter a value greater than 75 ($.75 * 2\text{GB} = 1.54 \text{GB}$). If you specify a value greater than 100 for this setting to initiate throttling based upon the specified number in MB, do not enter a value greater than 1536. If you are running a 64 bit version of BizTalk Server, specify a value of either 100 (100%) or 2048 (2GB) to initiate throttling when the maximum available process memory of 2 GB is used.

Adapters That Do Not Support 64-bit Hosts

The following adapters are not supported to run on 64-bit host instances:

- FTP adapter
- POP3 adapter

Make sure you run these adapters in a 32-bit host instance.

Configure the MIME/SMIME Encoder to run in 32-bit mode

In BizTalk Server 2010, the MIME/SMIME encoder pipeline component does not have native 64-bit support. This means that this component must be run in a 32-bit emulation mode process (WOW64). This implies that the host instance in which this encoder component (or the send pipeline of which it is a part) runs must be running in 32-bit emulation mode. Be aware of the performance (and other) implications of this restriction for other elements of BizTalk running in this same host instance.

Best Practices for BizTalk Server Settings

This topic lists best practices that you should follow as you perform operational readiness procedures for BizTalk Server.

Configure message batching to increase adapter performance

- Minimize the number of transactions performed by an adapter by combining more than one operation into a single batch.
- Limit the batch size based on the total number of bytes in the batch, in addition to message count. For more information about limiting the batch size, see [Configuring Batching to Improve Adapter Performance](#).

Adjust the large message threshold

- To improve throughput, increase the large message threshold, which lowers the number of large messages that are buffered to disk during mapping. For more information about increasing the threshold, see [How to Adjust the Message Size Threshold](#).

Determine the information you need to track during planning

- You should decide during the planning stages which information you need to track. This way, after you deploy the project, you can set the tracking options and limit the amount of tracked data to give you only the information you need.



Note

For more information about best practices related to tracking, see [Planning for Tracking](#) in this guide and [Health and Activity Tracking](#) (<http://go.microsoft.com/fwlink/?LinkId=154187>).

Do not track all messages

- We recommend that you not track all messages. This is because each time a message is touched, BizTalk Server makes another copy of the message. You can instead narrow the scope by tracking only a specific port. This helps to maximize the performance of your system and to keep the databases uncluttered.

Set tracking on send ports and receive ports instead of on a pipeline

- If you set tracking options on pipelines, you will also set the tracking options globally for every port that uses the pipeline. This in turn may result in far more data being tracked than you intend, which will slow system performance. Instead, you can set tracking options on send ports and receive ports.

Adjust throttling based on resource utilization

- Throttling in BizTalk Server is configured by default to provide good protection for the system. Monitor the performance counters for throttling states to see if throttling is taking place. Then gauge for yourself if the resource on which throttling is based (for example, database size or memory usage) is under or over utilized. Next, adjust the throttling thresholds up or down accordingly. For more information, see [Adjusting Throttling Thresholds: When and Why](http://go.microsoft.com/fwlink/?LinkId=154188) (http://go.microsoft.com/fwlink/?LinkId=154188).

Use the PassThruTransmit pipeline if possible

- If no document processing is required before sending a message to its destination, use the PassThruTransmit pipeline instead of the XML send pipeline.

Minimize usage of orchestration “Shape start and end” tracking events

- While orchestration shape tracking has obvious benefits for orchestration debugging, it has performance and scalability implications. The **Shape start and end** tracking event can cause significant overhead. It is best to minimize its usage in production environments where high throughput is necessary.



Note

Shape start and end tracking events are ENABLED by default on all orchestrations.

See Also

[Checklist: Configuring BizTalk Server](#)

Checklist: Providing High Availability with Fault Tolerance or Load Balancing

This topic lists steps that you should complete to configure fault tolerance and/or load balancing for the components of a production BizTalk Server environment to provide high availability. Configuration of fault tolerance for these components will allow the system to continue operating if a particular component fails.

Steps	Reference
Create multiple BizTalk host instances and implement host clustering support for those adapters that require it.	High Availability for BizTalk Hosts
Configure tracking hosts for high availability.	High Availability for BizTalk Hosts
Configure the BizTalk Server databases on a clustered SQL Server instance on a Windows server cluster.	High Availability for Databases

Steps	Reference
Configure tracking hosts for high availability. Verify that there are at least N+1 host instances with tracking enabled where N is the number of MessageBox databases in the BizTalk group.	To enable tracking for a host, select the option to Allow Host Tracking on the Host Properties dialog box available from the BizTalk Server Administration console. For more information about setting host properties, see the topic How to Modify Host Properties (http://go.microsoft.com/fwlink/?LinkId=154359).
Practice database failover regularly to ensure failover functionality.	This responsibility should be directly assigned to ensure that this is done on a consistent basis.
Verify there is enough processing power and memory on the passive cluster nodes to enable multiple SQL Server instances to run simultaneously in failover scenarios.	If two SQL instances regularly consume over 50 percent of the resources on individual cluster nodes, then when both instances fail over to a single node, each instance will incur degraded performance. Therefore, testing should be performed to ensure that a single cluster node will be able to handle the load until the failed node is brought back online. If a single node cannot handle the load of both SQL instances then consider adding additional cluster nodes to implement an Active/Active/Passive cluster topology.
Implement a storage area network (SAN) to house the BizTalk Server databases.  Note If possible, configure the SAN disks using RAID 1+0 (a stripe of mirror sets) topology for maximum performance and high availability.	In the "BizTalk Server Database Optimization white paper" (http://go.microsoft.com/fwlink/?linkid=104427), see the "Disk Infrastructure" section under "Performance Tuning".
Use Windows Clustering to cluster the Enterprise Single Sign-On (SSO) master secret server.	High Availability for the Master Secret Server  Note Do not cluster the SSO service on a computer running BizTalk Server unless you cluster SSO and a BizTalk host in the same cluster group. For more information about clustering the SSO service and a BizTalk Server host in the same cluster group see How to Cluster SSO and a BizTalk Host in the Same Cluster Group (http://go.microsoft.com/fwlink/?LinkId=154367).
Back up the Enterprise Single Sign-On	See How to Back Up the Master Secret

Steps	Reference
(SSO) Master Secret.	(http://go.microsoft.com/fwlink/?LinkID=151395).
Configure the Internet Information Services (IIS) Web server for isolated host instances and the BAM Portal Web page to be highly available using Network Load Balancing (NLB) or other load balancing device.	For Windows Server 2008: See Network Load Balancing Deployment Guide (http://go.microsoft.com/fwlink/?LinkId=153139).

See Also

[Providing High Availability](#)

Checklist: Increasing Availability with Disaster Recovery

This topic describes the steps that you should follow to increase availability of a production BizTalk Server environment using disaster recovery. Disaster recovery is typically implemented after providing high availability with fault tolerance and/or load balancing.

Backing Up BizTalk Server

Step	Reference
Keep a written record of all changes to your BizTalk Server system.	
Ensure that you have appropriate permissions to back up and restore BizTalk Server.	See Minimum Security User Rights (http://go.microsoft.com/fwlink/?LinkId=154374)
Create a highly available file share for purposes of storing the SQL Server logs. Configure a highly available file share at the hardware level using a SAN and/or at the Windows Server level by using Windows Clustering.	See How to create file shares on a cluster (http://support.microsoft.com/kb/224967)
Install and make available one or more standby SQL Server instances as the destination for BizTalk Server log shipping. The hardware for and the number of destination SQL Server instances should match the hardware for and number of SQL Server instances in the production environment. This will ensure that the destination SQL Server instance(s) can handle the same load as the production SQL	See Installing SQL Server 2008 (http://go.microsoft.com/fwlink/?LinkId=154377)

Step	Reference
Server instance(s).	
Prepare the disaster recovery site.	Prepare the Disaster Recovery Site
Back up and restore BizTalk Server databases	<ul style="list-style-type: none"> • Best Practices for Backing Up and Restoring Databases (http://go.microsoft.com/fwlink/?LinkId=157758) • Backing Up and Restoring BizTalk Server Databases (http://go.microsoft.com/fwlink/?LinkId=157757)
Configure BizTalk Server log shipping.	Configuring BizTalk Server Log Shipping
Configure the Backup BizTalk Server job.	How to Configure the Backup BizTalk Server Job (http://go.microsoft.com/fwlink/?LinkID=154072)
Configure the server where backups will be stored.	How to Configure the Destination System for Log Shipping (http://go.microsoft.com/fwlink/?LinkID=151402)
If you are using Business Activity Monitoring (BAM), back up the BAM databases.	Backing Up the BAM Analysis and Tracking Analysis Server Databases
If you are using BAM, back up the BAM portal application pools and virtual directories configuration information. If you are not using BAM, you do not need to perform this step.	How to Back Up the BAM Portal (http://go.microsoft.com/fwlink/?LinkId=154378)
Back up the BizTalk Server configuration file.	How to Back Up The BizTalk Server Configuration (http://go.microsoft.com/fwlink/?LinkId=154379)
If you are using Enterprise Single Sign-on, back up the master secret server.	How to Back Up the Master Secret (http://go.microsoft.com/fwlink/?LinkID=151395)

Restoring BizTalk Server

Steps	Reference
Restore the BizTalk group.	Restoring the BizTalk Group
Recover the run-time computers running BizTalk Server.	Recovering the Runtime Computers

Steps	Reference
Recover BAM alerts.	How to Recover BAM Alerts (http://go.microsoft.com/fwlink/?LinkId=154380)
Recover the BAM portal.	How to Recover the BAM Portal (http://go.microsoft.com/fwlink/?LinkId=154381)
Restore the master secret server.	How to Restore the Master Secret (http://go.microsoft.com/fwlink/?LinkId=151394)
Restore the BizTalk Server databases.	How to Restore Your Databases (http://go.microsoft.com/fwlink/?LinkId=151406)
Update references to the BAM database names.	<ul style="list-style-type: none"> • Updating References to the New BAM Primary Import Database • Updating References to the New BAM Archive Database • Updating References to the New BAM Star Schema Database • Updating References to the New BAM Analysis Database • Updating References to the New BAM Notification Services Databases • How to Resolve Incomplete Activity Instances (http://go.microsoft.com/fwlink/?LinkId=151475)

See Also

[Disaster Recovery](#)

Checklist: Monitoring Operational Readiness

This topic lists the steps that you should follow when monitoring a production BizTalk Server environment.

Steps	Reference
Select and implement a monitoring strategy for your BizTalk applications and infrastructure.	Monitoring the BizTalk Server Environment
Monitor disk space usage.	Monitoring Disk Space Usage
Monitor SQL Server: <ul style="list-style-type: none"> • Verify that computers running SQL Server 	<ul style="list-style-type: none"> • Monitoring SQL Servers • Monitoring BizTalk Server Databases

Steps	Reference
<p>housing the BizTalk Server databases are being monitored.</p> <ul style="list-style-type: none"> • Verify that the BizTalk Server jobs are being monitored. 	
<p>Monitor BizTalk applications:</p> <ul style="list-style-type: none"> • Modify existing rules and/or copy rules to a custom management pack to monitor a custom BizTalk application. • Create actions for each defined rule. • Create iterative processes to automate manual tasks. • Use threshold rules to automate manual tasks. 	<ul style="list-style-type: none"> • Monitoring Applications and Host Instances • Monitoring BizTalk Server

In This Section

- [Monitoring Disk Space Usage](#)

Monitoring Disk Space Usage

As part of the monitoring process for operational readiness of BizTalk Server, monitor the disk space usage as follows:

- **Determine the disk space required.**
When using File or MSMQ send / receive locations, ensure that there is ample disk space available to accommodate outages of BizTalk Server or of the receiving systems. For example, if BizTalk Server is writing files to a share on a SAN and the receiving system is down for two days, determine whether there is enough disk space to allow the files to queue up.
- **Clean up the BizTalk Server backup files directory periodically.**
You can perform this cleanup using a script called from a SQL Server Agent job.
- **Clean up the BizTalk Tracking database archive files directory periodically.**
- **Ensure that there is sufficient disk space available to accommodate larger BizTalk Server database (.mdf) and transaction log (.ldf) files during times of peak data flow.**

Monitoring BizTalk Server Databases

You can run the Monitor BizTalk Server SQL Agent job to identify any known issues in Management, Message Box, or DTA databases. The job is created when you configure a BizTalk group in BizTalk Server Administration console or upgrade BizTalk from the previous version.

The Monitor BizTalk Server Job

The Monitor BizTalk Server job scans for the following issues in Management, Message Box, and DTA databases:

**Note**

The Monitor BizTalk Server job only scans for issues. It does not fix the issues found.

- Messages without any references
- Messages without reference counts
- Messages with reference count less than 0
- Message references without spool rows
- Message references without instances
- Instance state without instances
- Instance subscriptions without corresponding instances
- Orphaned DTA service instances
- Orphaned DTA service instance exceptions
- TDDS is not running on any host instance when global tracking option is enabled

The Monitor BizTalk Server job is configured and automated to run once in a week. Since the job is computationally intensive, we recommended that you schedule it to run during periods of downtime or low traffic.

The job fails if it encounters any issues; the error string returned contains the number of issues found. Else, it runs successfully. You can see the details in the job history. If you run the job with Administrator privileges, the error string will be logged to both the job history and the SQL Server Application log.

**Important**

Failure of this job does not necessarily constitute a critical issue, but rather an issue that should be investigated and addressed as part of regular maintenance of the BizTalk Server databases. This job fails by design in the event it discovers one of the issues listed above.

Checklist: Maintaining and Troubleshooting BizTalk Server Databases

BizTalk Server databases and their health are very important for a successful BizTalk Server database messaging environment. This topic lists the steps that you must follow when maintaining or troubleshooting the BizTalk Server databases.

- [Maintaining BizTalk Server Databases](#)
- [Troubleshooting BizTalk Server Databases](#)

Maintaining BizTalk Server Databases

Steps	Reference
Disable the Auto Update Statistics and Auto Create	 Note These settings are done by default as part of the BizTalk Server configuration. You should not make changes to these settings.

Steps	Reference
<p>Statistics Options (applicable only to BizTalk Server MessageBox databases).</p>	<p>You must disable the Auto Create Statistics and the Auto Update Statistics options. To determine if these settings are disabled, execute the following stored procedures in SQL Server:</p> <pre>exec sp_dboption 'BizTalkMsgBoxDB', 'auto create statistics'</pre> <pre>exec sp_dboption 'BizTalkMsgBoxDB', 'auto update statistics'</pre> <p>The value returned for CurrentSetting should be OFF. If this value returned for CurrentSetting is ON, change it to OFF by executing the following stored procedures in SQL Server:</p> <pre>exec sp_dboption 'BizTalkMsgBoxDB', 'auto create statistics', 'off'</pre> <pre>exec sp_dboption 'BizTalkMsgBoxDB', 'auto update statistics', 'off'</pre> <p>For more information about these settings see the following Microsoft Knowledge Base articles:</p> <ul style="list-style-type: none"> • 917845—"You experience blocking, deadlock conditions, or other SQL Server issues when you try to connect to the BizTalkMsgBoxDB database in BizTalk Server" (http://go.microsoft.com/fwlink/?LinkId=153429). • 912262—"The auto update statistics option, the auto create statistics option, and the Parallelism setting are turned off in the SQL Server database instance that hosts the BizTalk Server BizTalkMsgBoxDB database" (http://go.microsoft.com/fwlink/?LinkId=153430).
<p>Set the Max Degree of Parallelism property</p>	<p> Note</p> <p>These settings are done by default as part of the BizTalk Server configuration. You should not make changes to these settings.</p> <p>Set the Max Degree of Parallelism run_value and config_value properties to a value of one (1) on the SQL Server instances that host the BizTalk Server Messagebox databases. To check the Max Degree of Parallelism setting, execute the following stored procedure against the Master database in SQL Server:</p> <pre>exec sp_configure 'max degree of parallelism'</pre> <p>If the run_value and config_value are not set to a value of one (1), execute the following stored procedure in SQL Server:</p> <pre>exec sp_configure 'max degree of parallelism', '1'</pre> <pre>reconfigure with override</pre> <p>For more information about how the Max Degree of Parallelism setting affects BizTalk Server, see the following Microsoft Knowledge Base articles:</p> <ul style="list-style-type: none"> • 899000—"The Parallelism setting for the instance of SQL Server when you configure BizTalk Server"

Steps	Reference
	<p>(http://go.microsoft.com/fwlink/?LinkId=153432).</p> <ul style="list-style-type: none"> • 917845—"You experience blocking, deadlock conditions, or other SQL Server issues when you try to connect to the BizTalkMsgBoxDb database in BizTalk Server" (http://go.microsoft.com/fwlink/?LinkId=153429).
<p>Determine when you can rebuild BizTalk Server indexes</p>	<p>Most of the indexes in BizTalk Server databases are clustered (index ID: 1). The DBCC SHOWCONTIG command can be used to display fragmentation information for tables in the BizTalk Server databases. These indexes are GUID-based so it is normal for fragmentation to occur. If the Scan Density value of DBCC SHOWCONTIG is less than 30%, the indexes can be rebuilt during downtime. Many tables in the BizTalk Server databases contain columns that use DataType definitions where online indexing cannot be done. Therefore, the indexes for tables in the BizTalk Server databases should never be rebuilt while BizTalk is processing data. For more information on how to rebuild the BizTalk indexes, see the following Microsoft Knowledge Base article:</p> <ul style="list-style-type: none"> • 917845—"You experience blocking, deadlock conditions, or other SQL Server issues when you try to connect to the BizTalkMsgBoxDb database in BizTalk Server" (http://go.microsoft.com/fwlink/?LinkId=153429). <p>For more information about index fragmentation and workload types, see the white paper Microsoft SQL Server 2000 Index Defragmentation Best Practices (http://go.microsoft.com/fwlink/?LinkId=101580).</p> <p> Note You can also use the sys.dm_db_index_physical_stats function to look for fragmentation information in SQL Server 2005 and SQL Server 2008 SP1. For more information, see sys.dm_db_index_physical_stats (Transact-SQL) (http://go.microsoft.com/fwlink/?LinkId=158493).</p>
<p>Monitor the database for locks, blocks, or deadlocks</p>	<p>It is an expected behavior for locks and blocks to occur on the SQL Server databases used by BizTalk Server. However, it is not expected to have these locks or blocks to continue for an extended period of time. Extended blocking and deadlocking on the SQL Server databases used by BizTalk Server are indicators of a potential problem. For the current known causes of deadlocking and blocking on the SQL Server databases used by BizTalk Server, review the following Microsoft Knowledge Base article:</p> <ul style="list-style-type: none"> • 917845—"You experience blocking, deadlock conditions, or other SQL Server issues when you try to connect to the BizTalkMsgBoxDb database in BizTalk Server"

Steps	Reference
	<p>(http://go.microsoft.com/fwlink/?LinkId=153429).</p>
<p>Monitor the size of databases and tables</p>	<p>The size of the BizTalk Server Messagebox database should typically be no more than approximately 5GB. The BizTalkMsgBoxDb database should not be holding any data, and should be considered a buffer until the data is processed or moved to the BizTalkDTADb database. An environment with a powerful SQL Server backend and numerous long running orchestrations may have a BizTalkMsgBoxDb database larger than 5GB. A high volume environment with no long-running orchestrations should have a BizTalk Server Messagebox database much smaller than 5GB. The BizTalk Server tracking database can vary greatly in size but if query performance decreases dramatically, then the tracking database is probably too large. As a rule of thumb, a BizTalk Server tracking database larger than 15-20 GB is considered too large and may adversely impact performance. The following issues may be attributable to BizTalk Server databases that are too large:</p> <ul style="list-style-type: none"> • The BizTalk Server Messagebox database continues to grow while the data size (not just the log file) remains large. BizTalk Server takes a longer time than normal to process even a simple message flow scenario. • Group Hub queries take a longer time than normal and may even timeout. • The database log file never gets truncated. • The BizTalk SQL Agent jobs run slower than normal. • Some tables are considerably large or have too many rows compared to normal. <p>The BizTalk Server databases can become large for several reasons including:</p> <ul style="list-style-type: none"> • BizTalk SQL Agent Jobs not running • Excessive suspended message or service instances • Disk failures • High levels of tracking • BizTalk Server throttling • Poor SQL Server performance • Network latency issues <p>Similarly, you can have too many rows in a table. There is no set number of rows that are too many. Additionally, this number of rows varies by what kind of data is stored in the table. For example, a <i>dta_DebugTrace</i> table that has more than 1 million rows probably has too many rows. A <i>HostNameQ_Suspended</i> table that has more than 200,000 rows probably</p>

Steps	Reference
	<p>has too many rows.</p> <p>Make sure that you know what is expected in your environment to determine whether a data issue is occurring.</p>
<p>Enable tracking on BizTalk Server host</p>	<p>By default, tracking is enabled on the default host. BizTalk requires the Allow Host Tracking option be checked on a single HOST. When tracking is enabled, the Tracking Data Decode Service (TDDS) moves the tracking event data from the BizTalk Server Messagebox database to the BizTalk Server tracking database. If no BizTalk Server hosts are configured with the option to Allow Host Tracking or if the tracking host is stopped, then TDDS will not run and the TrackingData_x_x tables in the BizTalk Server Messagebox database will grow unchecked. Therefore, a dedicated BizTalk Server host should be configured with the option to Allow Host Tracking. For more information about configuring a dedicated tracking host see Configuring a Dedicated Tracking Host.</p> <p> Note</p> <p>To allow TDDS to maintain new tracking events in high volume scenarios, you can create multiple instances of a single tracking host but no more than one host should be configured for tracking.</p>
<p>Use the correct BizTalk SQL Server Agent jobs</p>	<p>Execution of the BizTalk Server SQL Agent jobs are crucial for managing the BizTalk Server databases and for maintaining optimal performance.</p> <ul style="list-style-type: none"> • The Backup BizTalk Server SQL Server Agent job is the only supported method to back up the BizTalk Server databases. This job requires you to set up all BizTalk Server databases to use a Full Recovery Model. You should configure this job for a healthy BizTalk Server environment. You can use the SQL Server methods to back up the BizTalk Server databases only if the SQL Server service is stopped and if all BizTalk Server processes are stopped. <p> Note</p> <p>For more information about using the SQL Server full recovery model when configuring the SQL Agent Backup BizTalk Server job, see Log Shipping (http://go.microsoft.com/fwlink/?LinkId=153450) or Backup Under the Full Recovery Model (http://go.microsoft.com/fwlink/?LinkId=156509).</p> <ul style="list-style-type: none"> • The MessageBox_Message_ManageRefCountLog_BizTalkMsgBoxDb SQL Server Agent job is designed to run indefinitely. As a result the SQL Agent job history may not indicate that this SQL Agent job has successfully completed; this behavior is by design. If there is a failure,

Steps	Reference
	<p>the job will restart within 1 minute and continue running unabated. Therefore, failure notifications for this job can typically be ignored. If the job history for the <code>MessageBox_Message_ManageRefCountLog_BizTalkMsgBoxDb</code> SQL Server Agent job indicates that this job is constantly failing and restarting then further investigation into the cause of the failure/restart cycle may be required.</p> <ul style="list-style-type: none"> • The MessageBox_Message_Cleanup_BizTalkMsgBoxDb SQL Server Agent job is the only job that should not be manually enabled because it is initiated by the <code>MessageBox_Message_ManageRefCountLog_BizTalkMsgBoxDb</code> job. • The DTA Purge and Archive SQL Server Agent job maintains the BizTalk Server tracking database by purging and archiving tracked messages. This job reads every row in the table and compares the timestamp of each row to determine if the record should be removed. <p> Note When troubleshooting the BizTalk Server SQL Server Agent jobs, verify that all SQL Agent jobs except the <code>MessageBox_Message_ManageRefCountLog_BizTalkMsgBoxDb</code> are completing without errors.</p> <p>For more information about the BizTalk Server 2010 SQL Agent Jobs used in SQL Server:</p> <ul style="list-style-type: none"> • See Database Structure and Jobs (http://go.microsoft.com/fwlink/?LinkId=153451). • See Description of the SQL Server Agent jobs in BizTalk Server (http://go.microsoft.com/fwlink/?LinkId=153452).
Monitor and terminate suspended instances	<p>Service instances can be suspended (resumable) or suspended (not resumable). These service instances may be Messaging, Orchestration, or Port. BizTalk Server 2010 accommodates termination and removal of these instances by using the Group Hub page in the BizTalk Server Administration Console or through the use of the <code>Terminate.vbs</code> script. For more information about the <code>Terminate.vbs</code> script, see Removing Suspended Service Instance (http://go.microsoft.com/fwlink/?LinkId=153453).</p> <p> Tip You can also use the Terminator tool to remove suspended instances. The Terminator tool is available at http://go.microsoft.com/fwlink/?LinkId=151931. Use of this tool is not supported by Microsoft, and Microsoft makes no guarantees</p>

Steps	Reference
	<p>about the suitability of this programs. Use of this program is entirely at your own risk.</p> <p>The terms "orphans" and "zombies" are often used interchangeably. An orphaned or zombie message is a message that does not have an associated service instance, typically because the service instance has terminated before the message was received. An orphaned or zombie service is a service that does not have any associated messages. For more information about zombie messages and service instances in BizTalk Server see Zombies in BizTalk Server (http://go.microsoft.com/fwlink/?LinkId=153454).</p>
<p>Monitor the performance counters of the PhysicalDisk performance object</p>	<p>BizTalk Server makes a large number of short, very quick transactions to SQL Server within one minute. If the SQL Server cannot sustain this activity, you may experience BizTalk Server performance issues. Monitor the Avg. Disk sec/Read, Avg. Disk sec/Transfer, and Avg. Disk sec/Write performance monitor counters in the PhysicalDisk performance object. The optimal value is less than 10 ms (milliseconds). A value of 20 ms or larger is considered poor performance.</p> <ul style="list-style-type: none"> • For more information about SQL Server 2005 performance, see Troubleshooting Performance Problems in SQL Server 2005 (http://go.microsoft.com/fwlink/?LinkId=153585). • For more information about SQL Server 2008 performance, see Troubleshooting Performance Problems in SQL Server 2008 (http://go.microsoft.com/fwlink/?LinkId=153586). • For more information about BizTalk Server 2010 database high availability, see Providing High Availability for BizTalk Server Databases (http://go.microsoft.com/fwlink/?LinkId=153587). <p>You can also refer to the following Microsoft Knowledge Base articles for more information about SQL Server performance:</p> <ul style="list-style-type: none"> • 298475 – "How to troubleshoot SQL Server performance issues" (http://go.microsoft.com/fwlink/?LinkId=153588). • 271509 – "How to monitor blocking in SQL Server 2005" (http://go.microsoft.com/fwlink/?LinkId=153589).
<p>Follow best practices for BizTalk Server databases.</p>	<p>See Best Practices for Maintaining BizTalk Server Databases.</p>

Troubleshooting BizTalk Server Databases

Perform the following tasks to troubleshoot any issues with BizTalk Server databases.

Steps	Reference
<p>Ensure all required BizTalk SQL Server Agent jobs are enabled and running</p>	<p>All the BizTalk SQL Server Agent jobs except the MessageBox_Message_ManageRefCountLog_BizTalkMsgBoxDb job should be enabled and running successfully. Do not disable any other job.</p> <p>If a failure occurs, use the View History option in SQL Server to view the error information, and then troubleshoot the failure accordingly. Remember that the MessageBox_Message_ManageRefCountLog_BizTalkMsgBoxDb SQL Server Agent job runs infinitely. Therefore, you should only be concerned if the job history reports that the job constantly fails and restarts.</p>
<p>Use the MsgBoxViewer tool to analyze the BizTalk MessageBox and other databases</p>	<p> Important</p> <p>Use of this tool is not supported by Microsoft, and Microsoft makes no guarantees about the suitability of this programs. Use of this program is entirely at your own risk.</p> <p>The MsgBoxViewer tool is available from http://go.microsoft.com/fwlink/?LinkId=151930 (http://go.microsoft.com/fwlink/?LinkId=151930). The MsgBoxViewer tool is useful for troubleshooting because it provides an HTML report that has detailed information about table sizes and the row count. The report can also help determine whether BizTalk Server is throttling. Additionally, the tool provides a snapshot of the BizTalk Server databases and the BizTalk Server configuration.</p> <p>When BizTalk Server is running slower than usual, run the MsgBoxViewer tool, click to select all queries on the Optional Queries tab, and then review the generated HTML report for any problems. The Summary Report section lists warnings in yellow and potential problems in red.</p> <p>Additionally, you can use the MsgBoxViewer tool to determine which tables are the largest and have the most records. For a list of tables that typically grow the largest and for instructions on how to manage those tables, see Large-growing BizTalk Server Database Tables.</p>
<p>Use the Terminator tool to resolve issues, if any, identified by the MsgBoxViewer tool</p>	<p> Important</p> <p>Use of this tool is not supported by Microsoft, and Microsoft makes no guarantees about the suitability of this programs. Use of this program is entirely at your own risk.</p> <p>Run the Terminator tool available at Terminator (http://go.microsoft.com/fwlink/?LinkId=151931). This tool enables users to easily resolve any issues identified by the BizTalk MsgBoxViewer tool. For more information on how the Terminator tool</p>

Steps	Reference
	<p>integrates with the BizTalk MsgBoxViewer tool, see Using BizTalk Terminator to resolve issues identified by BizTalk MsgBoxViewer (http://go.microsoft.com/fwlink/?LinkId=151932).</p>
<p>Investigate deadlock scenarios</p>	<p>In a deadlock scenario, enable DBCC tracing on the SQL Server so that the deadlock information is written to the SQLERROR log.</p> <p>In SQL Server 2008 or SQL Server 2005, execute the following statement to enable DBCC tracing for deadlock scenarios:</p> <pre>DBCC TRACEON (1222,-1)</pre> <p>You can also use the PSSDIAG utility to collect data on the Lock:Deadlock event and the Lock:Deadlock Chain event. For more information about the PSSDIAG utility, see PSSDIAG data collection utility (http://go.microsoft.com/fwlink/?LinkId=153627).</p> <p>The BizTalkMsgBoxDB database is a high-volume and high-transaction Online Transaction Processing (OLTP) database. With such databases, some deadlocking is expected and this deadlocking is handled internally by the BizTalk Server engine. When this behavior occurs, no errors are listed in the error logs. When you investigate a deadlock scenario, the deadlock that you are investigating in the output must be correlated with a deadlock error in the event logs.</p>
<p>Look for blocked processes</p>	<p>You can use Activity Monitor in SQL Server to obtain the server process identifier (SPID) of a locking system process. You can then run the SQL Profiler to determine the SQL statement that is executing in the locking SPID. You can use the PSSDIAG utility to troubleshoot locking and blocking issues in SQL Server. The utility captures all the Transact-SQL events that have the blocking script enabled. For more information about the PSSDIAG utility, see PSSDIAG data collection utility (http://go.microsoft.com/fwlink/?LinkId=153627).</p> <p>In SQL Server 2005 and later versions, you can specify the blocked process threshold setting to determine which SPID or SPIDs are blocking longer than the threshold that you specify. For more information about the blocked process threshold option, see blocked process threshold Option (http://go.microsoft.com/fwlink/?LinkId=153628).</p> <p> Note When you experience a locking or blocking issue in SQL Server, we recommend that you contact Microsoft Customer Support Services. Microsoft Customer Support Services can</p>

Steps	Reference
	help you configure the correct PSSDiag utility options.
Delete all unwanted data	<p>If the databases have grown to become too large and if the data contained in the databases will not be required any longer, the preferred method is to delete the data.</p> <p> Caution Do not use this method in any environment where the data is business critical or if the data is needed.</p> <p> To purge the BizTalkMsgBox database</p> <ol style="list-style-type: none"> 1. Download the Msgbox_cleanup_logic.sql script from Microsoft Knowledge Base article 924715, FIX: Message data is not deleted from the tracking database after you run the bts_CleanupMsgbox stored procedure in a BizTalk Server 2006 test environment (http://go.microsoft.com/fwlink/?LinkId=153630). 2. Back up all BizTalk Server databases. 3. Copy the Msgbox_cleanup_logic.sql script to the computer hosting the SQL Server. 4. Execute this SQL script against the BizTalkMsgBoxDb database to update the bts_CleanupMsgbox stored procedure. 5. Stop all BizTalk hosts, services, and custom isolated adapters. If you use HTTP or the SOAP adapter, restart the IIS services. 6. Execute the bts_CleanupMsgbox stored procedure on all the BizTalkMsgBoxDb databases. <p> Caution Do not run the bts_CleanupMsgbox stored procedure on a production server that is running BizTalk Server. You should only run the bts_CleanupMsgbox stored procedure in a test environment. Running the bts_CleanupMsgbox stored procedure in a production environment is not supported.</p> <ol style="list-style-type: none"> 7. Restart all hosts and BizTalk Server services. <p> To purge the BizTalkDTADB database</p> <ul style="list-style-type: none"> • Method 1

Steps	Reference
	<ul style="list-style-type: none"> a. Back up all BizTalk Server databases. b. Execute the dtasp_PurgeAllCompletedTrackingData stored procedure. For more information about the stored procedure, see How to Manually Purge Data from the BizTalk Tracking Database (http://go.microsoft.com/fwlink/?LinkId=153635). <ul style="list-style-type: none"> • Method 2. Use this option only if the BizTalkDTADb database contains many incomplete instances that must be removed. <ul style="list-style-type: none"> a. Back up all BizTalk Server databases. b. Stop all BizTalk hosts, services, and custom isolated adapters. If you use HTTP or the SOAP adapter, restart the IIS services. c. Execute the dtasp_CleanHMData stored procedure on the BizTalkDTADb database. d. Restart all hosts and BizTalk Server services. <p>nNote</p> <p>If you must have the tracking data, back up the BizTalkDTADb database, restore the database to another SQL Server, and then purge the original BizTalkDTADb database.</p>

If you want help analyzing the MsgBoxViewer data or PSSDIAG output, contact Microsoft Customer Support Services. Before you contact Customer Support Services, compress the MsgBoxViewer data, the PSSDIAG output, and the updated event logs (.evt files). You may have to send these files to a BizTalk Server support engineer

In This Section

- [Best Practices for Maintaining BizTalk Server Databases](#)
- [Large-growing BizTalk Server Database Tables](#)
- [Tools and Utilities for Troubleshooting](#)

See Also

[SQL Server Settings That Should Not Be Changed](#)

Best Practices for Maintaining BizTalk Server Databases

This topic lists some best practices for maintaining BizTalk Server databases.

- Make sure the SQL Server Agent is running on the SQL Server. When the SQL Server Agent is stopped, the built-in BizTalk SQL Server Agent jobs that are responsible for database maintenance cannot run. This behavior causes database growth, and this growth may cause

performance issues. For information about monitoring SQL Server Agent Jobs see [Monitoring SQL Server Agent Jobs](#).

- Make sure the SQL Server LDF and MDF files are on separate drives. Having the LDF and MDF files for the BizTalkMsgBoxDb and BizTalkDTADb databases on the same drive may result in disk contention.
- Do not enable message body tracking, if not required. Frequently, you may want to enable message body tracking while you develop and troubleshoot a solution. If so, make sure you disable message body tracking when you are done. If you keep message body tracking enabled, the BizTalk Server databases grow. If you have a business need that requires you to enable message body tracking, confirm that the **TrackedMessages_Copy_BizTalkMsgBoxDb** and **DTA Purge and Archive** SQL Server Agent jobs are running successfully.
- Typically, smaller transaction logs cause better performance. To keep the transaction logs smaller, configure the **Backup BizTalk Server** SQL Server Agent job to run more frequently. For more information, see the [BizTalk Server Database Optimization white paper](#) (<http://go.microsoft.com/fwlink/?LinkId=153594>).
- Use the BizTalk Server Best Practices Analyzer (BPA) to evaluate an existing BizTalk Server deployment. The BPA performs numerous database-related checks. You can download the BizTalk Server Best Practices Analyzer tool from [BizTalk Server Best Practices Analyzer tool](#) (<http://go.microsoft.com/fwlink/?LinkId=83317>).

See Also

[Checklist: Maintaining and Troubleshooting BizTalk Server Databases](#)

[Large-growing BizTalk Server Database Tables](#)

Large-growing BizTalk Server Database Tables

The following table lists the BizTalk Server tables that typically grow the largest. You can use this data to determine where a potential problem may exist.

Table	Description	Comments
<i>HostNameQ_Suspended</i> table	This table contains a reference to messages in the Spool table that are associated with suspended instances for the particular host. This table is in the BizTalkMsgBoxDb database.	If the <i>HostNameQ_Suspended</i> tables have many records, the tables could be containing valid suspended instances that appear in the Group Hub page. You can terminate these instances. If these instances do not appear in the Group Hub , the instances are probably caching instances or orphaned routing failure reports. When you terminate suspended instances, you clean up the items in this table and their associated rows in the Spool and Instances tables.

Table	Description	Comments
<i>HostNameQ</i> table	This table contains a reference to messages in the Spool table that are associated with the particular host and are not suspended. This table is in the BizTalkMsgBoxDb database.	<p>If the <i>HostNameQ</i> tables have many records, the following kinds of instances may exist:</p> <ul style="list-style-type: none"> • Ready-to-run instances • Active instances • Dehydrated instances <p>BizTalk Server needs time to "catch up" and process the instances. This table can grow when the incoming rate of processing outpaces the outgoing rate of processing. This scenario can also occur due to large BizTalkDTADB database or SQL Server disk delays.</p>
Spool, Parts, and Fragments tables	These tables store actual message data in the BizTalkMsgBoxDb database.	The Spool, Parts, and Fragments tables having many records imply that there are a large number of messages are currently active, dehydrated, or suspended. Depending on the size, the number of parts, and the fragmentation settings in these tables, a single message may spawn all these tables. Each message has exactly one row in the Spool table and at least one row in the Parts table.
Instances table	This table stores all instances and their current status in the BizTalkMsgBoxDb database.	The BizTalk Server Administrator should not allow for many suspended instances to remain in the Instances table. Many dehydrated instances should only remain if the business logic requires long-running orchestrations. Remember that one service instance can be associated with many messages in the Spool table.
TrackingData_x_x table	This table stores the tracked events in the BizTalkMsgBoxDb database for Tracking Data Decode Service (TDDS) to move the events to the	If the TrackingData_x_x tables are large, either the TDDS is not running or is not running successfully. If the TDDS is running, review the event logs and the TDDS_FailedTrackingData table in the BizTalkDTADB database for error information.

Table	Description	Comments
	BizTalkDTADb database.	
Tracking_Fragmentsx, Tracking_Partsex, Tracking_Spoolx tables	Two of each of these tables is in the BizTalkMsgBoxDb and BizTalkDTADb databases. One is online, and the other is offline.	The TrackedMessages_Copy_BizTalkMsgBoxDb SQL Server Agent job moves tracked message bodies directly to these tables in the BizTalkDTADb database.
dta_ServiceInstances table	This table stores tracked events for service instances in the BizTalkDTADb database.	If this table is large, the BizTalkDTADb database is probably large.
dta_DebugTrace table	This table stores the Orchestration debugger events in the BizTalkDTADb database.	<p>If the dta_DebugTrace table has many records, orchestration shape tracking is being used or was being used. If orchestration debugging is not required for regular operations, disable orchestration shape tracking for all orchestrations. If orchestration shape tracking is already disabled and a backlog exists in the BizTalkMsgBoxDb database, the dta_DebugTrace table may continue to grow because TDDS continues to move this data into the dta_DebugTrace table. For instructions on how to disable tracking for an orchestration, see To disable tracking for an orchestration.</p> <p>To control the size of the BizTalkDTADb tracking database, you may choose to disable global tracking. For more information see How to Turn Off Global Tracking (http://go.microsoft.com/fwlink/?LinkId=153687). For more information about tracking database sizing guidelines, see Tracking Database Sizing Guidelines (http://go.microsoft.com/fwlink/?LinkId=15368).</p>

Table	Description	Comments
		8).
dta_MessageInOutEvents table	This table stores tracked event messages in the BizTalkDTADB database. These tracked event messages include message context information.	If the dta_DebugTrace table and the dta_MessageInOutEvents table in the BizTalkTrackingDb database are too large, you can truncate the tables manually after you stop the tracking host. For instructions on how to truncate the tables, follow the procedures under the section "dta_DebugTrace table" of Microsoft Knowledge Base article 952555, How to maintain and troubleshoot BizTalk Server databases (http://go.microsoft.com/fwlink/?LinkId=158847).
dta_ServiceInstanceExceptions table	This table stores error information for any suspended service instance in the BizTalkDTADB database.	The dta_ServiceInstanceExceptions table typically becomes large in an environment that regularly has suspended instances.

Tools and Utilities for Troubleshooting

This topic describes several tools and utilities that can be useful for diagnosing the root cause of a problem in a Microsoft BizTalk Server component or dependency.

Troubleshooting	Tool	Use
SQL Server Troubleshooting	SQL Activity Monitor	SQL Server 2008 Activity Monitor provides information about SQL Server processes and how these processes affect the current instance of SQL Server. For more information about SQL Server 2008 Activity Monitor, see Activity Monitor (http://go.microsoft.com/fwlink/?LinkId=146355) in the SQL Server documentation. For information about how to open Activity Monitor from SQL Server Management Studio, see How to: Open Activity Monitor

Troubleshooting	Tool	Use	
		(SQL Server Management Studio) (http://go.microsoft.com/fwlink/?LinkId=135094) in the SQL Server documentation.	
	SQL Server 2008 Data Collection	SQL Server 2008 provides a data collector that you can use to obtain and save data that is gathered from several sources. The data collector enables you to use data collection containers, which enable you to determine the scope and frequency of data collection on a computer that is running SQL Server 2008. For more information about implementing SQL Server 2008 data collection, see Data Collection (http://go.microsoft.com/fwlink/?LinkId=146356) in the SQL Server documentation.	
	Performance-related Troubleshooting	Relog	The Relog utility is used to extract performance counters from logs created by Performance Monitor and convert the data into other formats, such as tab-delimited text files (text-TSV), comma-delimited text files (text-CSV), binary files, and SQL databases.

Troubleshooting	Tool	Use	
			<p>This data can then be analyzed and queried using other tools, such as Log Parser, to generate statistics for key performance indicators (KPIs). The Relog utility is provided with Windows Server 2003 and subsequent versions.</p>
Windows Performance Analysis Tools		<p>Windows Performance Tools are designed for analysis of a wide range of performance problems including application start times, boot issues, deferred procedure calls and interrupt activity (DPCs and ISRs), system responsiveness issues, application resource usage, and interrupt storms. For more information, see Windows Performance Analysis Developer Center (http://go.microsoft.com/fwlink/?LinkId=139763).</p>	
Pathping		<p>Pathping provides information about possible data loss at one or more router hops on the way to a target host. To do so, pathping sends Internet Control Message Protocol (ICMP) packets to each router in the path. Pathping.exe is available with all versions of Windows since Windows 2000 Server.</p>	

Troubleshooting	Tool	Use
IOMeter		<p>IOMeter is an open source tool used for measuring disk I/O performance. For more information, see IOMeter (http://go.microsoft.com/fwlink/?LinkId=122412).</p> <p> Important Use of this tool is not supported by Microsoft, and Microsoft makes no guarantees about the suitability of this programs. Use of this program is entirely at your own risk.</p>
General troubleshooting	<ul style="list-style-type: none"> • Event Viewer • Network Monitor • SQL Server Profiler • SQL Server Query Editor • DTCTester • DTCPing • Performance Console • RegMon • FileMon • Debug Diagnostics Tool of the IIS Diagnostics Toolkit 	<p>See Tools and Utilities to Use for Troubleshooting (http://go.microsoft.com/fwlink/?LinkId=154416).</p>

See Also

[Tools for Testing](#)

[Checklist: Configuring BizTalk Server](#)

Checklist: Testing Operational Readiness

Testing for operational readiness is a vital procedure that is often overlooked or is performed only minimally. Thorough testing is a critical requirement of any enterprise-level application, and a solution developed using BizTalk Server is no exception. At a minimum, you should perform the following testing before moving a BizTalk solution into production:

Steps	Reference
Unit testing	Performing Unit Testing
Functional testing	Performing Functional Testing
Bottleneck testing and tuning	Performing Bottleneck Testing and Tuning
Load and maximum sustainable throughput (MST) testing	Performing Load and Throughput Testing
Availability testing: <ul style="list-style-type: none">• Test individual hardware component failure.• Test BizTalk Server failure.• Test cluster node failover.• Test recovery of BizTalk Server databases using SQL Server log shipping.	Performing Availability Testing

In This Section

- [Performing Unit Testing](#)
- [Performing Functional Testing](#)
- [Performing Bottleneck Testing and Tuning](#)
- [Performing Load and Throughput Testing](#)
- [Performing Availability Testing](#)
- [Tools for Testing](#)

Performing Unit Testing

Unit testing is focused at the component level and is basically a pass/fail test that verifies if individual components of the BizTalk solution perform as expected. You have several options for unit testing your BizTalk solution.

Using Visual Studio

Unit testing functionality is available with Visual Studio 2008 and later. For more information about the testing functionality that is available with Visual Studio, see [Testing the Application](#) (<http://go.microsoft.com/fwlink/?LinkId=159595>).

BizTalk Server 2010 also provides a unit testing feature to enable users to create unit tests for schemas, maps, and pipelines. For more information about this feature, see [Unit Testing with BizTalk Server Projects](#) (<http://go.microsoft.com/fwlink/?LinkId=158270>).

**Note**

Visual Studio is very useful for unit testing BizTalk artifacts such as orchestrations, schemas, pipelines, and pipeline components. BizTalk Server 2010 provides test classes that you can use with Visual Studio Team System to test BizTalk artifacts.

Using Non-Microsoft Tools

Two other commonly used tools for unit testing BizTalk solutions are **BizUnit** and **NUnit**. **BizUnit** works seamlessly with Visual Studio Team System Test Edition. Similarly, **NUnit** tests can be easily modified so that they can run as-is in Visual Studio Team System Test Edition. For more information about these tools, see [Tools for Testing](#).

**Note**

Use of **BizUnit** and **NUnit** are not supported by Microsoft, and Microsoft makes no guarantees about the suitability of these programs. Use of these programs is entirely at your own risk.

Using the BizTalk Server SDK

You can perform unit testing of individual BizTalk artifacts with utilities available in the BizTalk Server SDK. The table below provides a summary of the utilities in the SDK that can be used for unit testing:

Utility	Purpose
AS2 Sender Utility	Enables you to send an AS2 message to a Web site on a single computer. This utility simulates the sending of an AS2 message from a separate computer.
DSDump.exe	Enables you to dump the document schema structure, which is an in-memory lightweight representation of the one or more XSD schemas, with or without flat file annotations. This tool can be helpful when you get parsing engine errors such as \$Root\$0\$3\$2 and you need to decode them. Numbers after \$ mean 0-based index or records as they appear in the document schema.
FFAsm.exe	Runs the flat file assembler component, directly invoking it by emulating a send pipeline to enable you to see how it serializes or assembles a user's XML document(s) into a flat file document.
FFDasm.exe	Runs the flat file disassembler component, directly invoking it by emulating a receive

Utility	Purpose
	pipeline to enable you to see how it parses or disassembles a user's flat file document into one or more XML documents.
Pipeline.exe	Runs a send or receive pipeline; accepts one or more input documents and their parts, XSD schemas, and related information; and produces an output document after the pipeline runs. Pipeline.exe does not access BizTalk Server databases, so pipelines containing BizTalk Framework assembler and disassembler components that access BizTalk Server databases during execution may not be supported.
XMLAsm.exe	Runs the XML assembler component, directly invoking it by emulating a send pipeline to enable you to see how it serializes, assembles, or envelopes a user's XML document(s) into an output XML document.
XMLDasm.exe	Runs the XML disassembler component, directly invoking it by emulating a receive pipeline to enable you to see how it parses, disassembles, or un-envelopes a user's XML document into one or more XML documents.

For more information about the utilities available in the BizTalk Server SDK, see [Utilities in the SDK](http://go.microsoft.com/fwlink/?LinkId=154387) (<http://go.microsoft.com/fwlink/?LinkId=154387>).

See Also

[Tools for Testing](#)

Performing Functional Testing

You use functional testing to test a specific end-to-end scenario or a given use case in the context of a particular BizTalk application. A functional test should cover all the possible paths through a given scenario, including the failure paths. Failure paths should be evaluated to ensure that the application deals with failure conditions appropriately.

All the artifacts (such as orchestrations, custom pipeline components, and custom assemblies) should be invoked, and all the code branches through these objects should be tested as well. All the possible combinations of messages should be exercised to ensure that messages flow through the system correctly. Invalid messages should be tested as well to make sure that the

application reacts in the expected way in case of error and to test the code contained in all the exception blocks of orchestrations and custom components.

Automating Functional Testing

You should automate functional testing so that it is fast, so that it can be repeated, and so that it will avoid human errors. **BizUnit** is a declarative test framework designed to enable developers to rapidly design test cases. In fact, an XML configuration file called BizUnit XML test case is enough to define how a test should be performed. To run tests you can create your own custom driver or more easily leverage **Visual Studio Unit Testing** or **NUnit** to host and run your tests.

Every BizUnit XML test case contains three phases: **TestSetup**, **TestExecution**, and **TestCleanup**. Each of these phases can contain zero or more test steps. Each step represents a unit of work and is implemented as a .NET class designed to perform a certain task. This framework provides a rich set of components. If you need to realize specialized components to meet specific requirements, however, you can write your own custom test step components. For more information about these tools, see [Tools for Testing](#).



Note

Use of this tool is not supported by Microsoft, and Microsoft makes no guarantees about the suitability of this program. Use of this program is entirely at your own risk.

See Also

[Checklist: Testing Operational Readiness](#)

Performing Bottleneck Testing and Tuning

You should complete performance testing to determine bottlenecks in the system and to tune the system accordingly.

Testing a Subsystem

A best practice for identifying system bottlenecks is to run performance tests on subsets of the entire system, for example:

- Establish baseline performance parameters for external systems that send messages to or receive message from BizTalk Server.
- Enlist orchestrations, but do not start them. Drop messages into the inbound queues/file locations and let the inbound receive adapters drain the queue/file locations and publish messages into the MessageBox database. This allows you to isolate your receive ports to determine their maximum sustained input rate.
- Once the messages are pulled into the MessageBox database, stop the receive adapters, enable the orchestration processes and/or send adapters, and then measure the rate at which orchestrations and/or send adapters are processing messages.

Testing the End-to-End System

Testing of input and output rates as described in preceding section is an effective way to isolate performance of the application subsystem, although it does not describe end-to-end performance. You should also test end-to-end performance because some bottlenecks cannot be identified until multiple resources begin to contend for the same shared resource (for example, the MessageBox database).

To generate load against a BizTalk Server environment, consider using the Microsoft BizTalk LoadGen 2007 tool. For more information about the LoadGen 2007 tool see [Microsoft BizTalk LoadGen 2007](http://go.microsoft.com/fwlink/?LinkID=59841) (<http://go.microsoft.com/fwlink/?LinkID=59841>).

To generate and analyze a performance report for a BizTalk Server environment, consider using the Performance Analysis of Logs (PAL) tool. For more information about the PAL tool, see [Using the Performance Analysis of Logs \(PAL\) Tool](#).

What Developers, Operators, and Administrators Should Know

BizTalk Server developers should be well versed on BizTalk Server performance characteristics and tuning. Operators and administrators should be knowledgeable about MessageBox database scale-out aspects, SAN tuning, network tuning, and SQL Server database tuning (for example, see [SQL Server Settings That Should Not Be Changed](#)). Developers, operators, and administrators should be aware of how to tune BizTalk Server for high-throughput and low-latency.

Performing Load and Throughput Testing

You should make available an environment that matches your production environment for performance and stress testing. This environment should have all standard services installed and running, such as monitoring agents and antivirus software.

How Adding Applications Affects Load

You should also test new BizTalk applications alongside the other BizTalk applications that are going to run on the same hardware in production. This is because the new BizTalk applications put additional load on the computers running BizTalk Server and SQL Server. This is especially important in light of the host throttling algorithms that are used in BizTalk Server. The throttling algorithm monitors total available resources and hence the additional load incurred by a new BizTalk application may induce a throttling condition which affects all running BizTalk applications. For more information, see [How BizTalk Server Implements Host Throttling](#) (<http://go.microsoft.com/fwlink/?LinkId=154389>).

Testing Load and Determining Recovery Time

You should test all BizTalk applications for performance and stress before you put them into production. You should perform your testing against expected loads and against peak loads. You should determine the maximum sustainable throughput (MST) for the BizTalk application. In addition, you should determine how long it takes the system to recover from peak loads. If the system does not fully recover from a peak load before the next peak load occurs, then the system will get progressively farther behind and will not be able to fully recover. For more information, see:

- [Measuring Maximum Sustainable Engine Throughput](#) (<http://go.microsoft.com/fwlink/?LinkId=154388>)
- [Measuring Maximum Sustainable Tracking Throughput](#) (<http://go.microsoft.com/fwlink/?LinkID=153815>)

See Also

[Checklist: Testing Operational Readiness](#)

Performing Availability Testing

You should test your system for disaster recovery to verify its ability to recover from various levels of failure, ranging from small-scale failures (such as a network card failure) to the loss of a production server. Your disaster recovery testing should include the following steps:

- **Test individual hardware component failure.**

You should test the ability of the system to recover from an individual hardware component failure, such as a network or disk.

- **Test single BizTalk server failure.**

In most BizTalk Server production environments, host processing is spread out among multiple computers running BizTalk Server in a single BizTalk group. What is the ability of the BizTalk group to continue host processing in the event one of the servers in the group fails?

- **Test cluster node failover.**

If Windows Clustering is used to provide high availability for the BizTalk Server databases or BizTalk Hosts, you should verify cluster node failover functionality. For more information about using Windows Clustering to provide high availability for BizTalk Server, see [Checklist: Providing High Availability with Fault Tolerance or Load Balancing](#).

- **Test recovery of BizTalk Server databases using log shipping.**

You should verify the recovery of the BizTalk Server databases. For more information about using log shipping to back up and restore BizTalk Server databases, see [What Is BizTalk Server Log Shipping?](#) in this guide or [Log Shipping](#) (<http://go.microsoft.com/fwlink/?LinkID=153450>).

For a checklist of steps that you should follow to increase the availability of a BizTalk Server environment using disaster recovery, see [Checklist: Increasing Availability with Disaster Recovery](#).

See Also

[Increasing Availability for BizTalk Server](#)

Tools for Testing

The table below lists tools that you can use to perform the testing associated with operational readiness of BizTalk Server.

Testing	Tool	Use
Unit and Functional Testing	Microsoft Visual Studio	Used for unit testing .NET code. For more information about the testing functionality that is available with Visual Studio, see Testing the Application (http://go.microsoft.com/fwlink/?LinkId=159595).
	BizUnit	A framework designed for automated testing of BizTalk solutions.

Testing	Tool	Use
		 Note Use of this tool is not supported by Microsoft, and Microsoft makes no guarantees about the suitability of this programs. Use of this program is entirely at your own risk. For more information about BizUnit, see BizUnit - Framework for Automated Testing of Distributed Systems (http://go.microsoft.com/fwlink/?LinkID=85168).
	NUnit	Used for unit testing .NET code.  Note Use of this tool is not supported by Microsoft, and Microsoft makes no guarantees about the suitability of this programs. Use of this program is entirely at your own risk. For more information about NUnit, see Nunit (http://go.microsoft.com/fwlink/?LinkID=47931).
Code Coverage Testing	Visual Studio 2010 Code Coverage	Used to ensure that all execution paths through your application are adequately exercised, and to identify dead functions or classes in your code. In general, code that is unreachable or never executed should be removed. For more information about the code coverage feature of Microsoft Visual Studio 2010 see Using Code Coverage to Determine How Much Code Is Being Tested (http://go.microsoft.com/fwlink/?LinkId=210624).
Performance Testing	Performance Analysis of Logs (PAL) Tool	Tool for analyzing performance counter log files. For more information about the Performance Analysis of Logs (PAL) tool, see Performance Analysis of Logs (PAL) Tool (http://go.microsoft.com/fwlink/?LinkID=98098).
	Microsoft BizTalk LoadGen 2007 tool	Load generation tool used to run performance and stress tests against BizTalk Server. For more information about the Microsoft

Testing	Tool	Use
		BizTalk LoadGen 2007 tool, see Microsoft BizTalk LoadGen 2007 Tool (http://go.microsoft.com/fwlink/?LinkId=59841).
	Visual Studio 2010 load testing	Available with Visual Studio 2010 Ultimate edition, allows you to simulate load from up to 250 users by default, and over 250 users using Visual Studio Load Test Virtual User Pack. For more information about performing load testing using Visual Studio see Testing Application Performance and Stress (http://go.microsoft.com/fwlink/?LinkId=203129).
	BizTalk Server 2006 Orchestration Profiler	Used to view orchestration tracking data for a specified period of time; helpful for determining where performance bottlenecks exist in orchestrations.  Note Use of this tool is not supported by Microsoft, and Microsoft makes no guarantees about the suitability of this programs. Use of this program is entirely at your own risk. For more information about the BizTalk Server 2006 Orchestration Profiler, see BizTalk Server 2006 Orchestration Profiler (http://go.microsoft.com/fwlink/?LinkId=102209).
Web service testing	soapUI	An open source utility that can be used to test Web services.  Note Use of this tool is not supported by Microsoft, and Microsoft makes no guarantees about the suitability of this programs. Use of this program is entirely at your own risk. For more information about soapUI, see soapUI.org (http://go.microsoft.com/fwlink/?LinkId=102196).
	Fiddler	Useful for seeing HTTP traffic "on the wire." For more information about using the Fiddler Tool

Testing	Tool	Use
		<p>for debugging HTTP, see Fiddler PowerToy - Part 1: HTTP Debugging (http://go.microsoft.com/fwlink/?LinkID=84796).</p> <p> Note Use of this tool is not supported by Microsoft, and Microsoft makes no guarantees about the suitability of this programs. Use of this program is entirely at your own risk.</p> <p>The Fiddler tool can be downloaded from Introducing Fiddler (http://go.microsoft.com/fwlink/?LinkID=99357).</p>
Debugging	DebugView	<p>Tool for monitoring kernel-mode and Win32 debug output either locally or remotely. For more information about DebugView, see DebugView for Windows (http://go.microsoft.com/fwlink/?LinkId=102210).</p>
	BizTalk MsgBoxViewer	<p>Analyzes the BizTalk MessageBox and other databases and generates an HTML report containing warnings, if any, and other information related to the databases.</p> <p> Note Use of this tool is not supported by Microsoft, and Microsoft makes no guarantees about the suitability of this programs. Use of this program is entirely at your own risk.</p> <p>The BizTalk MsgBoxViewer tool can be downloaded from BizTalk MsgBoxViewer - download here the latest version of the tool (http://go.microsoft.com/fwlink/?LinkId=151930).</p>
	Terminator	<p>Resolves any issues identified by the BizTalk MsgBoxViewer tool. For more information on how the Terminator tool integrates with the BizTalk MsgBoxViewer tool, see Using BizTalk Terminator to resolve issues identified by BizTalk MsgBoxViewer (http://go.microsoft.com/fwlink/?LinkId=151932).</p>

Testing	Tool	Use
		 Note Use of this tool is not supported by Microsoft, and Microsoft makes no guarantees about the suitability of this programs. Use of this program is entirely at your own risk. The Terminator tool can be downloaded from Terminator (http://go.microsoft.com/fwlink/?LinkId=151931).
	BizTalk Server Best Practices Analyzer	The BizTalk Server Best Practices Analyzer examines a BizTalk Server deployment and generates a list of issues pertaining to best practices standards. The tool performs configuration-level verification by gathering data from different information sources, such as Windows Management Instrumentation (WMI) classes, SQL Server databases, and registry entries. The data is then used to evaluate the deployment configuration. The tool reads and reports only and does not modify any system settings, and is not a self-tuning tool. The BizTalk Server Best Practices Analyzer tool can be downloaded from BizTalk Server Best Practices Analyzer (http://go.microsoft.com/fwlink/?LinkId=83317).

See Also

[Checklist: Testing Operational Readiness](#)

Routine Maintenance Checklists

This section lists the maintenance checks that should be performed on a BizTalk Server system on a routine basis as part of a preventive maintenance program. Performing these checks will

help you determine whether there are reliability, administration, security, or performance issues with the system.

The topics in this section provide links to monitoring and maintenance topics, both inside and outside this document, for the maintenance checks.

In This Section

- [Checklist: Performing Daily Maintenance Checks](#)
- [Checklist: Performing Weekly Maintenance Checks](#)
- [Checklist: Performing Monthly Maintenance Checks](#)

Checklist: Performing Daily Maintenance Checks

This topic describes some of the items that you should check on a daily basis to help monitor the status of a BizTalk Server system. You must perform most of these checks consistently and archive the results over a period of time to obtain the greatest benefit. We recommend that you automate routine maintenance checks whenever possible.

Steps	Reference
Check for failed disks in the hardware RAID (reliability check).	Manage Disks (http://go.microsoft.com/fwlink/?LinkId=158666).
Check for messages requiring manual intervention such as suspended messages (reliability check).	For information about manually checking for suspended messages, see Investigating Orchestration, Port, and Message Failures (http://go.microsoft.com/fwlink/?LinkId=154512).
Check for any errors or issues with the various databases associated with BizTalk Server, especially the MessageBox database.	Run the BizTalk MsgBoxViewer tool available from BizTalk MsgBoxViewer - download here the latest version of the tool (http://go.microsoft.com/fwlink/?LinkId=151930). This tool analyzes the BizTalk MessageBox and other databases and generates an HTML report containing warnings, if any and other information related to the databases.  Tip You can also save the reports for later use. These reports might be useful when troubleshooting issues with the BizTalk application.  Note Use of this tool is not supported by Microsoft, and Microsoft makes no

Steps	Reference
	<p>guarantees about the suitability of this programs. Use of this program is entirely at your own risk.</p>
<p>Resolve issues, if any, identified by the BizTalk MsgBoxViewer tool.</p>	<p>Run the Terminator tool available at Terminator (http://go.microsoft.com/fwlink/?LinkId=151931). This tool enables users to easily resolve any issues identified by the BizTalk MsgBoxViewer tool. For more information about how the Terminator tool integrates with the BizTalk MsgBoxViewer tool, see Using BizTalk Terminator to resolve issues identified by BizTalk MsgBoxViewer (http://go.microsoft.com/fwlink/?LinkId=151932).</p> <p> Note Use of this tool is not supported by Microsoft, and Microsoft makes no guarantees about the suitability of this programs. Use of this program is entirely at your own risk.</p>
<p>Check the event logs for errors and warnings (administration check).</p>	<p>BizTalk Server 2010 errors and warning events are saved in the application log. The event source is "BizTalk Server 2010."</p>

See Also

[Maintaining Reliability](#)

[Monitoring BizTalk Server](#)

Checklist: Performing Weekly Maintenance Checks

This topic describes the steps involved in performing weekly maintenance checks of the reliability, administration, security, and performance of a BizTalk Server system.

Steps	Reference
<p>Ensure that each host has an instance running on at least two physical BizTalk servers (reliability check).</p>	<p>High Availability for BizTalk Hosts</p>
<p>Ensure that each receive location is</p>	<p>Scaling Out Receiving Hosts</p>

Steps	Reference
redundant (reliability check).	
Ensure that the SQL Server Agent service is running on the SQL server (administration check).	<ul style="list-style-type: none"> • How to: Start SQL Server Agent (http://go.microsoft.com/fwlink/?LinkId=154672). • SQL Server Agent (http://go.microsoft.com/fwlink/?LinkId=106728).
Ensure that all SQL Server jobs related to BizTalk Server are working properly (administration check).	<p>Monitoring SQL Server Agent Jobs</p> <p>If the SQL Server Agent jobs are not running, system performance will degrade over time. For more information about the SQL Server Agent jobs that BizTalk Server provides to help manage the BizTalk Server databases, see Database Structure and Jobs (http://go.microsoft.com/fwlink/?LinkId=153451).</p>
Ensure that the SQL Server jobs responsible for backing up BizTalk Server databases are running normally (administration check).	<ul style="list-style-type: none"> • How to Configure the Backup BizTalk Server Job (http://go.microsoft.com/fwlink/?LinkId=153813) • How to Schedule the Backup BizTalk Server Job (http://go.microsoft.com/fwlink/?LinkId=154674)
Ensure that the latest security updates are installed (security check).	Microsoft Update site at http://update.microsoft.com/microsoftupdate/v6/default.aspx
Analyze weekly performance monitoring logs against baseline and thresholds (performance check).	<ul style="list-style-type: none"> • Using the Performance Analysis of Logs (PAL) Tool • Troubleshooting Performance Issues
Ensure that the system is not experiencing frequent auto-growth of BizTalk Server databases (performance check).	<ul style="list-style-type: none"> • Defining Auto-Growth Settings for Databases (http://go.microsoft.com/fwlink/?LinkId=154677). • Tracking Database Sizing Guidelines (http://go.microsoft.com/fwlink/?LinkId=154678). • Identifying Bottlenecks in the Database Tier (http://go.microsoft.com/fwlink/?LinkId=101579). • Database File Initialization (http://go.microsoft.com/fwlink/?LinkId=101579). • Performing SQL Server Maintenance Procedures
Run SQL Server Profiler during high load to check for long response times and high resource usage (performance check).	Using SQL Server Profiler (http://go.microsoft.com/fwlink/?LinkId=106720).
Ensure that message batching for all adapters is appropriate for	<ul style="list-style-type: none"> • Configuring Batching to Improve Adapter Performance • How to Design a Performant Adapter

Steps	Reference
resource consumption or latency (performance check).	(http://go.microsoft.com/fwlink/?LinkId=154679).
Ensure that the large message threshold is appropriate for resource consumption (performance check).	<ul style="list-style-type: none"> • How to Adjust the Message Size Threshold • How BizTalk Server Processes Large Messages (http://go.microsoft.com/fwlink/?LinkId=154680).
Archive backup files and specify appropriate computers for backup	<p>To avoid potential data loss, you should specify a computer for your backup that is different from the computer with the original data, and for <destination path> you should specify a computer to store the database logs that is different from the computer with the original database logs.</p> <p>For more information about best practices for backup, see Best Practices for Backing Up and Restoring Databases (http://go.microsoft.com/fwlink/?LinkId=151391).</p>

See Also

[Routine Monitoring Tasks](#)

Checklist: Performing Monthly Maintenance Checks

This topic describes the steps involved in performing monthly maintenance checks of the reliability, administration, security, and performance of a BizTalk Server system.

Steps	Reference
Ensure the master secret key is backed up and readily available on offline storage (reliability check).	How to Back Up the Master Secret (http://go.microsoft.com/fwlink/?LinkId=151395).
Ensure that failover for all clustered services has been tested (reliability check).	Reviewing and Testing SQL Server Cluster Configuration for Failover Scenarios Testing Group Failover
Ensure that the Enterprise SSO service is clustered (reliability check).	Clustering the Master Secret Server
Ensure that the BizTalk Server databases are clustered under SQL Server services (reliability check).	Clustering the BizTalk Server Databases
Ensure that at least two physical BizTalk servers are part of the BizTalk group (reliability	Ensuring Multiple Servers Are Part of a BizTalk Group

Steps	Reference
check).	
Determine whether any unstable code is being used, and if so, use separate hosts (reliability check).	High Availability for BizTalk Hosts
Perform functional testing of all new BizTalk applications (reliability check).	<ul style="list-style-type: none"> • Testing an Application • Staging Tasks for BizTalk Application Deployment (http://go.microsoft.com/fwlink/?LinkId=154686).
Configure and schedule backup BizTalk Server jobs (reliability check).	<ul style="list-style-type: none"> • How to Configure the Backup BizTalk Server Job (http://go.microsoft.com/fwlink/?LinkID=153813) • How to Schedule the Backup BizTalk Server Job (http://go.microsoft.com/fwlink/?LinkId=154674)
Ensure that the correct version of a set of assemblies is installed on each BizTalk machine (integrity check).	<p>Use the BizTalk Assembly Checker and Remote GAC tool (BTSAssemblyChecker.exe) to check the versions of assemblies deployed to the BizTalk Management database and to verify that they are correctly registered in the GAC on all BizTalk Server computers. You can use this tool to verify that all the assemblies containing the artifacts of a certain BizTalk application are installed on all BizTalk nodes. The tool is particularly useful in conjunction with a solid versioning strategy to verify that the correct version of a set of assemblies is installed on each BizTalk machine, especially when side-by-side deployment approach is used. The tool is available with the BizTalk Server 2010 installation media at Support\Tools\x86\BTSAssemblyChecker.exe.</p>
Determine whether there are any unnecessary BizTalk applications, artifacts, and configurations (administration check).	<ul style="list-style-type: none"> • Remove all unnecessary BizTalk applications, artifacts, and configurations. • For more information about removing a BizTalk application or artifact using the BTSTask command-line tool see RemoveApp Command

Steps	Reference
	<p>(http://go.microsoft.com/fwlink/?LinkId=154687).</p> <ul style="list-style-type: none"> For more information about removing an artifact from an application using either the BizTalk Server Administration console or the BTSTask command-line tool, see How to Remove an Artifact from an Application (http://go.microsoft.com/fwlink/?LinkId=154688).
Check the BizTalk Server Administration console for any non-approved changes (administration check).	<p>Using the BizTalk Server Administration Console (http://go.microsoft.com/fwlink/?LinkId=154689).</p>
Check BTSNTSvc.exe.config for any non-approved modifications (administration check).	<p>BTSNTSvc.exe.config File (http://go.microsoft.com/fwlink/?LinkId=154690).</p>
Check the BizTalk Server-related registry keys for any non-approved modifications (administration check).	<p>Microsoft Knowledge Base article 256986, "Windows registry information for advanced users" (http://go.microsoft.com/fwlink/?LinkId=158859).</p>
Run the Best Practices Analyzer for BizTalk Server (administration check).	<p>BizTalk Server Best Practices Analyzer (http://go.microsoft.com/fwlink/?LinkId=83317).</p>
Ensure that the latest service packs and updates are installed (administration and security check).	<p>Microsoft Update (http://go.microsoft.com/fwlink/?LinkId=154691).</p>
Ensure that the artifacts for different trading partners are not installed on the same host (security check).	<p>Configuring Hosts and Host Instances</p>
Ensure that BizTalk Server is using only domain-level users and groups (security check).	<p>Domain Groups (http://go.microsoft.com/fwlink/?LinkId=154692).</p>
Ensure that the MSDTC Security Configuration is enabled (security check).	<p>Follow guidelines in the "Set the appropriate MSDTC Security Configuration options on Windows Server 2003 SP1, Windows XP SP2, Windows Server 2008, and Windows Vista" section of Troubleshooting Problems with MSDTC (http://go.microsoft.com/fwlink/?LinkId=154693).</p>
Determine whether the BizTalk Server cache refresh interval needs to be increased	<p>How to Adjust the Configuration Cache Refresh</p>

Steps	Reference
(performance check).	Interval
Determine whether the throttling options of each host need to be adjusted (performance check).	<ul style="list-style-type: none"> • For information about inbound and outbound host throttling, see What is Host Throttling? (http://go.microsoft.com/fwlink/?LinkId=154694). • For information about triggers, actions, and mitigation strategies for inbound and outbound throttling, see the “Throttling condition triggers, actions, and mitigation strategies” section of How BizTalk Server Implements Host Throttling (http://go.microsoft.com/fwlink/?LinkId=154695).
Determine whether unnecessary tracking is enabled, such as orchestration, shape, and Business Rule Engine (BRE) event tracking (performance check).	<ul style="list-style-type: none"> • How to Disable Tracking • Planning for Tracking • Best Practices for Tracking
Determine whether you are using a dedicated host for tracking maintenance (performance check).	Configuring a Dedicated Tracking Host
Check the BizTalk Server database sizes for an increasing trend (performance check).	<ul style="list-style-type: none"> • For more information about sizing the tracking database, see Tracking Database Sizing Guidelines (http://go.microsoft.com/fwlink/?LinkId=154677). • For more information about sizing the MessageBox, BizTalkDTADb, and BAMPrimaryImport databases, see Identifying Bottlenecks in the Database Tier (http://go.microsoft.com/fwlink/?LinkId=154678).

See Also

[Routine Maintenance Checklists](#)

Routine Performance Checklists

This section lists the performance checks that should be performed on a BizTalk Server system on a routine basis as part of a preventive maintenance program. Performing these checks will

help you determine whether there are any bottlenecks in your application setup or any performance issues with the system.

The topics in this section provide links to performance related topics, both inside and outside this document, for the performance checks.



Note

For a detailed discussion on performance related to BizTalk Server 2010 and related products, see the [BizTalk Server Performance Optimization Guide](http://go.microsoft.com/fwlink/?LinkID=150492) (<http://go.microsoft.com/fwlink/?LinkID=150492>).

In This Section

- [Checklist: Performing Weekly Performance Checks](#)
- [Checklist: Performing Monthly Performance Checks](#)

Checklist: Performing Weekly Performance Checks

This topic lists best practices that you should follow on a weekly basis when to avoid performance issues with a BizTalk Server system.

Steps	Reference
Set database auto-growth to a fixed number	<ul style="list-style-type: none">• Database auto-growth should be set to a fixed number of megabytes instead of a percentage, especially for the MessageBox and Tracking (DTA) databases. Depending on your BizTalk Server application and throughput, the MessageBox and Tracking databases can get quite large. If auto-growth is set to a percentage, then the auto-growth can be substantial as well.• If the system is new and the static sizes have not been definitively established, then configure files with the Enable Autogrowth option and specify file growth in Megabytes. The growth increment should generally be no larger than 100 MB (for large files), 10 MB (for medium-sized files), or 1 MB (for small files). See the Appendix B – Recommended BizTalk Server Database Configuration section of the BizTalk Server Database Optimization white paper (http://go.microsoft.com/fwlink/?LinkID=153594) for a table with suggested file sizes for each of the BizTalk Server databases.

Steps	Reference
Limit the events that you monitor with SQL Server Profiler	Use SQL Server Profiler to monitor only the events in which you are interested. If traces become too large, you can filter them based on the information you want, so that only a subset of the event data is collected. Monitoring too many events adds overhead to the server and the monitoring process, and can cause the trace file or trace table to grow very large, especially when the monitoring process takes place over a long period of time.
Configure message batching to increase adapter performance	<ul style="list-style-type: none"> • Minimize the number of transactions performed by an adapter by combining more than one operation into a single batch. • Limit the batch size based on the total number of bytes in the batch, in addition to message count. For more information about limiting the batch size, see Configuring Batching to Improve Adapter Performance.
Adjust the large message threshold	To improve throughput, increase the large message threshold, which lowers the number of large messages that are buffered to disk during mapping. For more information about increasing the threshold, see How to Adjust the Message Size Threshold .
Investigate memory leaks or out-of-memory exceptions in the BizTalk Server process	Memory leaks in BizTalk processes can occur due to varied reasons. See Microsoft Knowledge Base Article 918643, How to troubleshoot a memory leak or an out-of-memory exception in the BizTalk Server process (http://go.microsoft.com/fwlink/?LinkId=157212) for information about the scenarios in which a memory leak may occur, and how to fix it.

See Also

[Routine Performance Checklists](#)

[Checklist: Performing Monthly Performance Checks](#)

Checklist: Performing Monthly Performance Checks

This topic lists best practices that you should follow on a monthly basis to avoid performance issues with a BizTalk Server system.

Steps	Reference
Determine the information you need to track during planning	<p>You should decide during the planning stages which information you need to track, so that after you deploy the project you can set the tracking options and limit the amount of tracked data to give you only the information you need.</p> <p> Note For more information about best practices related to tracking, see Planning for Tracking in this guide and Health and Activity Tracking (http://go.microsoft.com/fwlink/?LinkId=154187) in the BizTalk Server documentation.</p>
Do not track all messages	<p>We recommend that you not track all messages, because each time a message is touched, BizTalk Server makes another copy. You can instead narrow the scope by tracking only a specific port. This helps to maximize the performance of your system and to keep the databases uncluttered.</p>
Do not track all events for orchestrations	<p>Tracking all events for an orchestration might increase the size of dta_DebugTrace and dta_MessageInoutEvents tables. For instructions on how to disable tracking for an orchestration, see To disable tracking for an orchestration.</p>
Set tracking on send ports and receive ports instead of on a pipeline	<p>If you set tracking options on pipelines, you will also set the tracking options globally for every port that uses the pipeline. This in turn may result in far more data being tracked than you intend, which will slow system performance. Instead, you can set tracking options on send ports and receive ports.</p>
Adjust throttling based on resource utilization	<p>Throttling in BizTalk Server is configured by default to provide good protection for the system. Monitor the performance counters for throttling states to see if throttling is taking place, then gauge for yourself if the resource on which throttling is based (for example, database size or memory usage) is under or overutilized, and then adjust the throttling thresholds up</p>

Steps	Reference
	<p>or down accordingly. For more information, see Adjusting Throttling Thresholds: When and Why (http://go.microsoft.com/fwlink/?LinkId=154188).</p>
<p>Use the PassThruTransmit pipeline if possible</p>	<p>If no document processing is required before sending a message to its destination, use the PassThruTransmit pipeline instead of the XML send pipeline.</p>
<p>Take into account various factors when you size the BizTalk Tracking database</p>	<ul style="list-style-type: none"> • When sizing the BizTalk Tracking database, account for SQL Server factors, such as index size, by adding a contingency multiplier to your calculations. • When determining the size of messages in the BizTalk Tracking database, add the average size of the message context to the message size if it is significant compared to the message size. • To limit the size of messages in the BizTalk Tracking database, limit the number of properties that you promote. • If the orchestration debugger option is enabled, take into account that the status of each shape in the orchestration is saved in the BizTalk Tracking database.
<p>Apply hardware solutions to avoid disk contention</p>	<p>To avoid disk contention in the MessageBox database, do the following:</p> <ul style="list-style-type: none"> • Use high-speed disks • Deploy the databases on a high-speed SAN • Separate the MessageBox database onto a dedicated server that is separate from the tracking databases • Scale up the CPUs and add more CPUs to the dedicated MessageBox database server • Move the PageFile and/or MSDTC log to a separate drive <p>For more information about avoiding database contention, see How to Avoid Disk Contention (http://go.microsoft.com/fwlink/?LinkId=158809).</p>

See Also

[Routine Performance Checklists](#)

[Checklist: Performing Weekly Performance Checks](#)

Checklists for Other Important Tasks

This section contains checklists for other important tasks that you might need to perform on an ad-hoc basis. These include such tasks as deploying an application in BizTalk Server, exporting bindings for an application, and updating an application.

In This Section

- [Checklist: Deploying an Application](#)
- [Checklist: Exporting Bindings from One Application to Another](#)
- [Checklist: Updating an Assembly](#)
- [Checklist: Updating an Application Using Side-by-Side Versioning](#)
- [Checklist: Updating an Orchestration Using Side-by-Side Versioning](#)
- [Checklist: Updating Artifacts in a BizTalk Application](#)
- [Checklist: Installing and Configuring Certificates](#)
- [Checklist: Monitoring BizTalk Server with Operations Manager 2007](#)
- [Checklist: Monitoring SQL Servers](#)
- [Checklist: Planning for Operations in a Secure Environment](#)

Checklist: Deploying an Application

This topic describes the steps involved in deploying a BizTalk application and its artifacts in a production environment. It demonstrates how to deploy an application in the development environment, export it into an .msi file, and then import it into the production environment from the .msi file.

Application deployment consists of importing the application into a group and installing the application on individual servers (host instances) in the group. For more information about importing and installing applications, see [How to Import an Application from an .msi File](#).

For more information about application deployment, see [The Application Deployment Process](#) (<http://go.microsoft.com/fwlink/?LinkId=154716>).

Steps	Reference
Ensure that you have the appropriate permissions to perform the deployment.	Permissions for Managing an Application
Set the deployment properties in Visual Studio for each project in the BizTalk Server solution, including the application that the assembly will be deployed to. Deploy (or redeploy) the required BizTalk Server assemblies into the BizTalk application in the development environment.	<ul style="list-style-type: none">• How to Set Deployment Properties in Visual Studio (http://go.microsoft.com/fwlink/?LinkId=154718).• Deploying BizTalk Assemblies from Visual Studio into a BizTalk Application (http://go.microsoft.com/fwlink/?LinkId=154719).

Steps	Reference
	<ul style="list-style-type: none"> • How to Redeploy a BizTalk Assembly from Visual Studio (http://go.microsoft.com/fwlink/?LinkId=154720). • Deploying an Assembly • "Deploying a BizTalk Application", "Creating a BizTalk Application", and "Deploying a BizTalk Assembly" sections of Best Practices for Deploying an Application. • "Deploying a BizTalk Application" section of Known Issues with Deploying an Application.
<p>Add artifacts to the BizTalk application and configure the artifacts in the development environment.</p> <p>This can include factoring artifacts into multiple BizTalk Applications.</p>	<ul style="list-style-type: none"> • How to Create or Add an Artifact (http://go.microsoft.com/fwlink/?LinkId=154724). • Managing Artifacts (http://go.microsoft.com/fwlink/?LinkId=154725). • Binding Files and Application Deployment (http://go.microsoft.com/fwlink/?LinkId=154726). • Adding Artifacts to an Application • "Adding Artifacts to a BizTalk Application" section of Best Practices in BizTalk Application Deployment • "Adding Artifacts to a BizTalk Application" section of Known Issues with Deploying an Application.
<p>Export the BizTalk application into an .msi file in the development environment.</p>	<ul style="list-style-type: none"> • How to Export an Application to an .msi File • "Exporting a BizTalk Application" section of Best Practices for Deploying an Application. • "Exporting a BizTalk Application" section of Known Issues with Deploying an Application.
<p>Import the BizTalk application into the production environment from the .msi file. (This operation consists of importing the application into the group and if the application includes file-based artifacts, installing the application on each server in the group.)</p>	<ul style="list-style-type: none"> • How to Export an Application to an .msi File • How to Install an Application (http://go.microsoft.com/fwlink/?LinkId=154728). • "Importing a BizTalk Application" section of Best Practices for Deploying an Application

Steps	Reference
	<ul style="list-style-type: none"> • "Importing a BizTalk Application" section of Known Issues with Deploying an Application
<p>Make any additional modifications to the BizTalk application in the production environment so it runs in that environment, such as changing port configurations. If you do so, export the application into an .msi file.</p>	<ul style="list-style-type: none"> • How to Export an Application to an .msi File • Creating and Modifying BizTalk Applications (http://go.microsoft.com/fwlink/?LinkId=154727). • How to Add a Binding File to an Application • "Exporting a BizTalk Application" section of Best Practices for Deploying an Application • "Exporting a BizTalk Application" section of Known Issues with Deploying an Application
<p>If you have made any additional modifications to the BizTalk application in the production environment, install the BizTalk application from the .msi file into any other computers in the production environment that will run it.</p> <p>Also, import the application into any other BizTalk groups where you want to deploy it, and if the application includes file-based artifacts, install the application on the servers in those groups.</p>	<ul style="list-style-type: none"> • How to Install a BizTalk Application (http://go.microsoft.com/fwlink/?LinkId=154728). • How to Import an Application from an .msi File • "Importing a BizTalk Application" section of Best Practices for Deploying an Application • "Importing a BizTalk Application" section of Known Issues with Deploying an Application
<p>Start the BizTalk application in the production environment from the BizTalk Server Administration console and verify that it is functioning correctly.</p>	<ul style="list-style-type: none"> • How to Start and Stop a BizTalk Application (http://go.microsoft.com/fwlink/?LinkId=154729). • Testing an Application

See Also

- [Deploying an Assembly](#)
- [Adding Artifacts to an Application](#)
- [How to Export an Application to an .msi File](#)
- [How to Export Bindings to a Binding File](#)
- [How to Import an Application from an .msi File](#)
- [How to Install an Application](#)
- [How to Import Bindings from a Binding File](#)

Checklist: Exporting Bindings from One Application to Another

This topic describes the steps involved in transferring the bindings of one application to another application in either a development or production environment. This process is similar to the process of deploying an application using an .msi file. However, when you deploy an application using an .msi file, the process will automatically create the application. When you transfer the bindings from one application to another, on the other hand, the destination application must already exist.

Steps	Reference
Ensure that you have the appropriate permissions to perform the export operation.	Permissions for Managing an Application
Create a destination application to import the application bindings into.	<ul style="list-style-type: none"> • Creating an Application • "Creating a BizTalk Application" section of Best Practices for Deploying an Application
Export the bindings of a source application into a binding file.	<ul style="list-style-type: none"> • How to Export Bindings to a Binding File • "Exporting a BizTalk Application" section of Best Practices for Deploying an Application • "Exporting a BizTalk Application" section of Known Issues with Deploying an Application.
Import the bindings in a binding file to the destination application.	<ul style="list-style-type: none"> • How to Import Bindings from a Binding File • "Importing a BizTalk Application" section of Best Practices for Deploying an Application • "Importing a BizTalk Application" section of Known Issues with Deploying an Application.

Checklist: Updating an Assembly

The following checklist describes the process of updating one or more artifacts in an application that has already been deployed, and then redeploying the application.



Note

If you cannot schedule downtime or have very long-running instances that cannot be terminated, update using side-by-side versioning.

Steps	Reference
Review the important considerations for updating artifacts in an application.	<ul style="list-style-type: none"> • Important Considerations for Updating Applications (http://go.microsoft.com/fwlink/?LinkId=1548)

Steps	Reference
	23). <ul style="list-style-type: none"> • Best Practices for Updating Applications
Ensure that you have appropriate permissions to perform the deployment.	Permissions for Managing an Application
Make any necessary changes to your assemblies, adding, removing, or reconfiguring artifacts as required. Deploy the assemblies from Visual Studio into a BizTalk application in the development environment.	<ul style="list-style-type: none"> • How to Update an Assembly • Updating an Artifact • "Updating an Artifact" and "Updating an Assembly" sections of Best Practices for Updating Applications • How to Deploy a BizTalk Assembly from Visual Studio (http://go.microsoft.com/fwlink/?LinkId=154824).
Test any new or changed artifacts, ensuring that any artifacts that may depend on the new or changed artifact are also tested.  Note When testing, be sure to consider dependencies that may exist between this application and other applications.	Testing Tasks for BizTalk Application Deployment (http://go.microsoft.com/fwlink/?LinkId=154825).
In the BizTalk Server Administration console, add, remove, or reconfigure artifacts in the application as necessary.	<ul style="list-style-type: none"> • How to Create or Add an Artifact (http://go.microsoft.com/fwlink/?LinkId=154724). • How to Remove an Artifact from an Application (http://go.microsoft.com/fwlink/?LinkId=154688). • Managing Artifacts (http://go.microsoft.com/fwlink/?LinkId=154725). • Updating an Artifact • "Updating an Artifact" section of Best Practices for Updating Applications.
Export the application containing the new or changed artifacts into an .msi file.	<ul style="list-style-type: none"> • Exporting BizTalk Applications, Bindings, and Policies (http://go.microsoft.com/fwlink/?LinkId=154826). • How to Export an Application to an .msi File

Steps	Reference
	<ul style="list-style-type: none"> • "Exporting a BizTalk Application" section of Best Practices for Deploying an Application.
<p>If the update will interfere with the application as it runs, schedule downtime, and stop the application that you want to update.</p>	<ul style="list-style-type: none"> • How to Start and Stop a BizTalk Application (http://go.microsoft.com/fwlink/?LinkID=154729). • "Starting or Stopping an Application" section of Best Practices for Updating Applications.
<p>Import the changed or updated artifacts from the .msi file into the application that you want to update, installing the application.</p> <p> Note</p> <p>When you update a BizTalk assembly, you should stop and unenlist artifacts before importing from the .msi file. You should re-enlist and then start BizTalk artifacts after you import from the .msi.</p>	<ul style="list-style-type: none"> • How to Import a BizTalk Application (http://go.microsoft.com/fwlink/?LinkID=154827). • How to Import an Application from an .msi File • "Importing a BizTalk Application" section of Best Practices for Deploying an Application. • How to Install a BizTalk Application (http://go.microsoft.com/fwlink/?LinkID=154728). • How to Install an Assembly in the GAC (http://go.microsoft.com/fwlink/?LinkID=154828).
<p>Start the application, resuming message publication. Restart all BizTalk host instances.</p>	<ul style="list-style-type: none"> • How to Start and Stop a BizTalk Application (http://go.microsoft.com/fwlink/?LinkID=154729).
<p>After importing an assembly that contains an orchestration, if the application to which you are importing it already contains an assembly that has the same name, public key token, and version, stop and start the host instances of the host to which the orchestration is bound. This ensures that the new version of the assembly is used by BizTalk Server.</p>	<ul style="list-style-type: none"> • How to Stop a Host Instance (http://go.microsoft.com/fwlink/?LinkID=154829). • How to Start a Host Instance (http://go.microsoft.com/fwlink/?LinkID=154830).

Checklist: Updating an Application Using Side-by-Side Versioning

The following checklist describes the process of deploying an updated version of a BizTalk application that will run side-by-side with an existing version.

A side-by-side installation enables you to roll out an application upgrade incrementally. You can make the installation available to a subset of business partners initially, rather than to all partners at once. Using this approach allows you to continue running the existing application to service the

users who are not yet using the new version until you are ready to completely move over to the new version.

The two side-by-side applications will have to receive messages on two different receive locations. As a result, to run side-by-side applications you must ask those trading partners who should use the new version of the application to send messages to the new receive location, so that they will be processed by the new version. Those trading partners who should use the old version should send messages to the previous receive location.

Steps	Reference
Create and implement a versioning strategy.	"Versioning" section in Best Practices for Updating Applications .
Make any necessary changes to the projects that you want to deploy into the new version of the application.	<ul style="list-style-type: none"> • How to Create or Add an Artifact (http://go.microsoft.com/fwlink/?LinkID=154724). • Managing Artifacts (http://go.microsoft.com/fwlink/?LinkID=154725). • Binding Files and Application Deployment (http://go.microsoft.com/fwlink/?LinkID=154726). • Adding Artifacts to an Application
Increment the version number of each assembly.	How to Update an Assembly
Set the deployment properties for each project in the solution (setting the destination application and enabling installation into the global assembly cache (GAC)).	How to Update an Assembly
Deploy solutions containing the changed assemblies into an application in your development environment.	<ul style="list-style-type: none"> • Deploying BizTalk Assemblies from Visual Studio into a BizTalk Application (http://go.microsoft.com/fwlink/?LinkID=154719). • How to Redeploy a BizTalk Assembly from Visual Studio (http://go.microsoft.com/fwlink/?LinkID=154720). • Deploying an Assembly
Create a new receive port and any needed receive locations specifying the new URLs that you want partners to send messages to. Create the appropriate send ports as	<ul style="list-style-type: none"> • How to Create a Receive Port (http://go.microsoft.com/fwlink/?LinkID=154843). • How to Create a Receive Location

Steps	Reference
<p>necessary.</p> <p>If necessary, bind the new application to the newly created receive and send ports, and test that the application works.</p>	<p>(http://go.microsoft.com/fwlink/?LinkId=154844).</p> <ul style="list-style-type: none"> • How to Create a Send Port (http://go.microsoft.com/fwlink/?LinkId=154845). • How to Configure an Application (http://go.microsoft.com/fwlink/?LinkId=154847).
<p>Export the new application into an .msi file from your development environment.</p>	<ul style="list-style-type: none"> • How to Export a BizTalk Application (http://go.microsoft.com/fwlink/?LinkId=154848). • Binding Files and Application Deployment (http://go.microsoft.com/fwlink/?LinkId=154726). • How to Export an Application to an .msi File • How to Export Bindings to a Binding File
<p>Import the application .msi file into the BizTalk group in your production environment.</p>	<ul style="list-style-type: none"> • How to Import a BizTalk Application (http://go.microsoft.com/fwlink/?LinkId=154827). • How to Install a BizTalk Application (http://go.microsoft.com/fwlink/?LinkId=154728). • How to Import an Application from an .msi File • How to Import Bindings from a Binding File
<p>Perform a full start of the application.</p>	<ul style="list-style-type: none"> • How to Start and Stop a BizTalk Application (http://go.microsoft.com/fwlink/?LinkId=154729). • Testing Tasks for BizTalk Application Deployment (http://go.microsoft.com/fwlink/?LinkId=154825).
<p>Notify your partners that they should start sending messages to the new URLs. Once they do this, the application begins processing messages for the specified partners.</p>	<p>-</p>

Checklist: Updating an Orchestration Using Side-by-Side Versioning

Changes to orchestrations can be more involved than changes to other artifacts, such as maps. If you have short-lived orchestrations, then a simple update may be sufficient. But if you have long-running orchestrations or cannot terminate existing instances, then side-by-side versioning will be your only option.

When an orchestration handles long-running transactions, you cannot change to the updated version of the orchestration immediately. You must allow the original version to finish processing its messages so that they are not lost. To accomplish this, you deploy the updated orchestration into the same application as the original one. You then stop the original version and start the updated version so that it receives all new messages while the previous version continues to process any in-flight messages. After the original orchestration has finished processing all of its messages, you undeploy it from the BizTalk application in which it was deployed.

Steps	Reference
After making the necessary changes to the orchestration, increment the assembly version number.	How to Update an Assembly
Deploy the assembly from Visual Studio into a BizTalk application, and then test the assembly.  Note Make sure you select the deployment option to install the assembly in the GAC.	Deploying BizTalk Assemblies from Visual Studio into a BizTalk Application (http://go.microsoft.com/fwlink/?LinkID=154719).
Export the assembly from the application in your test environment into an .msi file.	How to Export an Application to an .msi File
Import the .msi file into the BizTalk application in your production environment that contains the orchestration that you want to update.  Note You can use the following steps for testing the assembly as well as deploying it into your production environment.	How to Import an Application from an .msi File
Bind the updated orchestration using the same bindings as the original orchestration.	How to Configure Bindings for an Orchestration (http://go.microsoft.com/fwlink/?LinkID=154850).
Unenlist the original orchestration, and then start the updated orchestration.	For more information about deploying the orchestration programmatically, see Deploying

Steps	Reference
<p> Note</p> <p>To avoid any dropped messages, you should do this programmatically.</p>	<p>and Starting a New Version of an Orchestration Programmatically (http://go.microsoft.com/fwlink/?LinkId=154851).</p> <p>For more information about deploying the orchestration manually, see the following in BizTalk Server 2010 Help:</p> <ul style="list-style-type: none"> • How to Unenlist an Orchestration (http://go.microsoft.com/fwlink/?LinkId=154852). • How to Enlist an Orchestration (http://go.microsoft.com/fwlink/?LinkId=154853). • How to Start an Orchestration (http://go.microsoft.com/fwlink/?LinkId=154854).
<p>Monitor the system for instances of the original orchestration version using the Group Hub page query view.</p>	<p>How to View Instance Information for an Orchestration (http://go.microsoft.com/fwlink/?LinkId=154855).</p>
<p>When all of its active, dehydrated, and suspended instances are complete, undeploy the original orchestration from the application.</p>	<p>How to Remove an Orchestration from an Application (http://go.microsoft.com/fwlink/?LinkId=154856).</p>
<p>Optionally uninstall the original assembly version from the GAC on each computer running the application.</p>	<p>How to Uninstall an Assembly from the GAC (http://go.microsoft.com/fwlink/?LinkId=154857).</p>

Binding to Receive Ports and Locations

If you want to create new receive ports and locations for the new version of the orchestration, simply binding to the new ports and enlisting/starting the new artifacts will typically be sufficient. Creating new receive locations and ports is usually the preferred approach, especially if your scenario uses long-running orchestrations where a number of correlating receives still need to be processed. In this case, you may not be able to reuse existing receive ports or perform unenlistment. If you create new ports, be sure that it is possible for your backend and partner systems to handle this change. If not, you will have to wait for all long-running instances to culminate before upgrading.

If you want to use existing ports, do the following:

1. Bind the new version of the orchestration to existing ports.
2. Unenlist (but do not stop) the old orchestration version.
3. Enlist and start the new orchestration version.

**Note**

You can use a script to do steps 2 and 3 in one transaction, so that messages will not be missing subscriptions between manual clicking.

Checklist: Updating Artifacts in a BizTalk Application

For a checklist of tasks for updating artifacts in a BizTalk application, see [Checklist: Update the Artifacts in a BizTalk Application](http://go.microsoft.com/fwlink/?LinkId=155647) (http://go.microsoft.com/fwlink/?LinkId=155647).

See Also

[Checklists for Other Important Tasks](#)

Checklist: Installing and Configuring Certificates

This topic describes the steps involved in setting up certificates for use with BizTalk Server.

Steps	Reference
Assess your certificate requirements, and come to an agreement with your trading partner about your mutual responsibilities with respect to certificates.	"Assess and Plan Your Use of Certificates" section in Best Practices for Managing Certificates
<p>If you will be using the private key, request a private-public key pair for digital signatures from the certification authority (CA), and send the public key to your trading partner.</p> <p>If you will be using the public key, have your trading partner request a private-public key pair for digital signatures from the CA, and send the public key to you.</p>	How to Install Certificates for BizTalk Server
Install the private or public key into the appropriate certificate store, and have your trading partner do the same.	<ul style="list-style-type: none"> • How to Install Certificates for BizTalk Server • "Install Certificates" section of Best Practices for Managing Certificates
If using the certificate for MIME/SMIME messages, configure BizTalk Server to use the certificate.	<ul style="list-style-type: none"> • Configuring Certificates for MIME/SMIME Messages • "Configure BizTalk Server to Use Certificates for MIME/SMIME" section in Best Practices for Managing Certificates
(Optional) If using the certificate with an adapter, configure the adapter to use the certificate.	<ul style="list-style-type: none"> • Configuring Certificates with Adapters • "Configure a BizTalk Adapter to Use Certificates" section of Best Practices for

Steps	Reference
	Managing Certificates
(Optional) If using the certificate to perform party resolution, configure the party to use the certificate.	How to Configure Certificates for Party Resolution
(Optional) If exporting the certificate from one BizTalk group to another, add the certificate to a BizTalk application; export the application to an .msi file; and then import the application into a different BizTalk group.	<ul style="list-style-type: none"> • How to Configure Certificates for Party Resolution • How to Export an Application to an .msi File • How to Import an Application from an .msi File • "Exporting a Certificate from One BizTalk Group to Another" section of Best Practices for Managing Certificates

See Also

[Best Practices for Managing Certificates](#)

[Known Issues with Certificates in BizTalk Server](#)

Checklist: Monitoring BizTalk Server with Operations Manager 2007

This topic lists the high-level steps you can follow when preparing to monitor your BizTalk Server environment.

Step	Reference
Ensure that you have appropriate permissions to install and configure software on your BizTalk Server computer(s).	Minimum Security User Rights (http://go.microsoft.com/fwlink/?LinkID=154374)
Install Operations Manager 2007 agent on each BizTalk Server computer you want to monitor and point it to the Operations Manager 2007 server.	Refer to Deploying Operations Manager 2007 (http://go.microsoft.com/fwlink/?LinkId=110030)
Download and import the appropriate version of the management packs for the following: <ul style="list-style-type: none"> • BizTalk Server (required) • Enterprise Single Sign-On (required) • Windows Base OS (Server) (optional) • Microsoft Windows Server Cluster (if clusters are used, optional) • SQL Server 2008, SQL Server 2005 	<ul style="list-style-type: none"> • Download the Management Packs from System Center Management Pack Catalog (http://go.microsoft.com/fwlink/?LinkId=203227). • Import the management pack by following instructions at How to Import a Management Pack in Operations Manager 2007 (http://go.microsoft.com/fwlink/?LinkID=983)

Step	Reference
(optional) <ul style="list-style-type: none"> Internet Information Services (IIS) 2008, IIS 2003 (optional) Message Queuing (MSMQ) 3.0 (optional) 	48).
Read the best practices for using Operations Manager 2010 to monitor BizTalk Server.	Best Practices for Monitoring with Operations Manager 2007
Enable or disable the BizTalk Server management pack rules as appropriate.	Best Practices for Monitoring with Operations Manager 2007
Add any standalone Enterprise Single Sign-On (SSO) computers to the list of computers to be monitored by the BizTalk Server management pack.	How to Add Enterprise Single Sign-On Computers to the List of Computers Monitored by the BizTalk Server Management Pack (http://go.microsoft.com/fwlink/?LinkId=157263).

See Also

[Monitoring BizTalk Server with System Center Operations Manager 2007](#)

Checklist: Monitoring SQL Servers

This topic describes the steps that should be followed to monitor SQL Server in a production BizTalk Server environment.

Task	Reference
Install the Microsoft SQL Server Management Pack for Operations Manager 2007	SQL Server Monitoring Management Pack (http://go.microsoft.com/fwlink/?linkid=109858)
Designate the SQL Server databases as critical in the SQL Server Management Pack.	How to Mark BizTalk Server Databases for Customized Monitoring
Ensure that the SQL server machines are monitored.	Monitoring SQL Server Agent Jobs and Databases
Ensure that the BizTalk SQL Agent jobs are monitored.	Monitoring SQL Server Agent Jobs and Databases

Checklist: Planning for Operations in a Secure Environment

Running BizTalk Server in a secure environment requires additional steps for deployment and configuration. While default operating system installations need not take these into account, but scenarios where restrictive security policies have been applied, you should take into account

information in this section. The level of restriction applied onto servers may vary but information below should cover most cases and would be a a good starting point.

- [Security Considerations for Computers Running BizTalk Server](#)
- [Security Considerations for Computers Running SQL Server](#)
- [Additional Security Considerations](#)

Security Considerations for Computers Running BizTalk Server

The following table suggests the security-related settings on computers running BizTalk Server.

User Rights Assignment

To start the User Rights Assignment MMC Snap-in, click **Start**, click **Administrative Tools**, and then click **Local Security Policy**. In the **Local Security Policy** MMC snap-in, expand **Security Settings**, expand **Local Policies**, and then click **User Rights Assignment**.

Policy setting	Values	Reference and details
Log on as a service	BizTalk Application Users	Required to run BizTalk Host Instances. For more information about different user accounts, see Windows Groups and User Accounts in BizTalk Server (http://go.microsoft.com/fwlink/?LinkID=155755).
Log on as a service	RuleEngine Update Service Account	Required to run RuleEngine Update Service. For more information about different user accounts, see Windows Groups and User Accounts in BizTalk Server (http://go.microsoft.com/fwlink/?LinkID=155755).
Log on as a service	SSO Service Account	Required to run Enterprise Single Sign-On Service. For more information about different user accounts, see Windows Groups and User Accounts in BizTalk Server (http://go.microsoft.com/fwlink/?LinkID=155755).

System Services

To start the Services MMC Snap-in, click **Start**, click **Run**, and in the **Run** dialog box, type **services.msc** and press ENTER.

Service name	Startup type ¹	Details	User ²	Permissions	Details
COM+ System Application	Automatic	Required by BizTalk to run properly	(default)		
DHCP Client	Automatic	Required even if	(default)		

Service name	Startup type ¹	Details	User ²	Permissions	Details
		IP addresses are static			
Distributed Transaction Coordinator	Automatic	Required by BizTalk to run properly	SSO Service Account	Full Control	Required to start SSO service
			BizTalk Hosts Service Account	Full Control	Required to start BizTalk Hosts
			Network Service	Full Control	Required by IIS
HTTP SSL ³	Automatic	Required by IIS	(default)		
IPSEC Services ³	Automatic	IPSEC increases network security if used	(default)		
Netlogon	(default)		Local Service	Full Control	
NT LM Security Support Provider ³	Automatic	Required for Kerberos Authentication for BizTalk Server in SQL	(default)		
Remote Access Connection Manager	(default)		SSO Service Account	Full Control	Required to start SSO service
			BizTalk Hosts Service Account	Full Control	Required to start BizTalk Hosts
			Network Service	Full Control	Required by IIS
Remote Procedure Call (RPC) Locator	Automatic	Required by BizTalk	(default)		
WinHTTP Web Proxy Auto-	(default)		SSO Service	Full Control	Required to start SSO

Service name	Startup type ¹	Details	User ²	Permissions	Details
Discovery Service			Account		service
			BizTalk Hosts Service Account	Full Control	Required to start BizTalk Hosts

¹ A value of (default) means that the default settings applied by the security policy are not changed

² A value of (default) means that the default user permissions for the service have not been changed

Registry Settings

To start the Registry Editor, click **Start**, click **Run**, and in the **Run** dialog box, type **regedit** and press ENTER.

Key	User	Permissions	Details
HKLM\SYSTEM\CurrentControlSet\Services\DHCP	Network Service	Full Control	Required by DHCP Client Service
HKLM\SYSTEM\CurrentControlSet\Services\TCPIP	Network Service	Full Control	Required by DHCP Client Service

Security Considerations for Computers Running SQL Server

The following table suggests the security-related settings on computers running SQL Server.

User Rights Assignment

To start the User Rights Assignment MMC Snap-in, click **Start**, click **Administrative Tools**, and then click **Local Security Policy**. In the **Local Security Policy** MMC snap-in, expand **Security Settings**, expand **Local Policies**, and then click **User Rights Assignment**.

Policy setting	Values	Reference and details
Act as part of the operating system	SQL Server Agent Service Account, SQL Server Service Account	Required to run SQL Server. For more information see Setting Up Windows Service Accounts (http://go.microsoft.com/fwlink/?LinkId=157415).
Adjust memory quotas for a process	SQL Server Agent Service Account, SQL	Required to run SQL Server. For more information see Setting Up Windows Service Accounts

Policy setting	Values	Reference and details
	Server Service Account	(http://go.microsoft.com/fwlink/?LinkId=157415).
Bypass traverse checking	SQL Server Agent Service Account, SQL Server Service Account	Required to run SQL Server. For more information see Setting Up Windows Service Accounts (http://go.microsoft.com/fwlink/?LinkId=157415).
Create global objects	SQL Server Service Account	Required by SSIS service. For more information see Setting Up Windows Service Accounts (http://go.microsoft.com/fwlink/?LinkId=157415).
Enable computer and user accounts to be trusted for delegation	SQL Server Service Account, SQL Server Servers, BizTalk Server Servers, SQL Server Cluster Name	Required by BizTalk Server. Server name is in the form <servername>\$. For more information, see How to: Enable Kerberos Authentication on a SQL Server Failover Cluster (http://go.microsoft.com/fwlink/?LinkId=157417).
Log on as a service	SQL Server Agent Service Account, SQL Server Service Account	Required to run SQL Server. For more information see Setting Up Windows Service Accounts (http://go.microsoft.com/fwlink/?LinkId=157415).
Log on as a service	SSO Service Account	Required to run Enterprise Single Sign-On Service. For more information about different user accounts, see Windows Groups and User Accounts in BizTalk Server (http://go.microsoft.com/fwlink/?LinkId=157415).
Log on as batch job	SQL Server Agent Service Account, SQL Server Service Account	Required to run SQL Server. For more information see Setting Up Windows Service Accounts (http://go.microsoft.com/fwlink/?LinkId=157415).
Replace a process level token	SQL Server Agent Service Account, SQL Server Service Account	Required to run SQL Server. For more information see Setting Up Windows Service Accounts (http://go.microsoft.com/fwlink/?LinkId=157415).

System Services

To start the Services MMC Snap-in, click **Start**, click **Run**, and in the **Run** dialog box, type **services.msc** and press ENTER.

Service name	Startup type ¹	Details	User ²	Permissions	Details
DHCP Client	Automatic	Required even if	(default)		

Service name	Startup type ¹	Details	User ²	Permissions	Details
		IP addresses are static			
Distributed Transaction Coordinator	Manual	Service startup managed by Cluster Service	SSO Service Account	Full Control	Required to start SSO service
			Network Service	Full Control	Required by IIS
HTTP SSL ³	Automatic	Required by IIS	(default)		
IPSEC Services ³	Automatic	IPSEC increases network security if used	(default)		
Netlogon	(default)		Local Service	Full Control	
NT LM Security Support Provider ³	Automatic	Required for Kerberos Authentication for BizTalk Server in SQL	(default)		
Remote Access Connection Manager	(default)		SSO Service Account	Full Control	Required to start SSO service
			Network Service	Full Control	Required by IIS
Server	Automatic	Used for Clustered File Share resources	Network Service	Full Control	
WinHTTP Web Proxy Auto-Discovery Service	(default)		SSO Service Account	Full Control	Required to start SSO service
	World Wide Web Publishing Service	Automatic	Required by SQL Server Reporting Services	(default)	

¹ A value of (default) means that the default settings applied by the security policy are not changed

² A value of (default) means that the default user permissions for the service have not been changed

Registry Settings

To start the Registry Editor, click **Start**, click **Run**, and in the **Run** dialog box, type **regedit** and press ENTER.

Key	User	Permissions	Details
HKLM\SYSTEM\CurrentControlSet\Services\DHCP	Network Service	Full Control	Required by DHCP Client Service
HKLM\SYSTEM\CurrentControlSet\Services\TCPIP	Network Service	Full Control	Required by DHCP Client Service

Additional Security Considerations

The following table suggests the other important security-related settings for your BizTalk Server environment.

Affected artifact	Change	Reference and details
SSO Service Account	Grant Full Control Permission on Cluster in Cluster Manager	This change is required for SSO in order to work properly
SQL Server Service Account, SQL Server Servers, BizTalk Server Servers, SQL Server Cluster Name	Trust for Delegation in Active Directory	Required for proper Kerberos authentication. For more information, see How to: Enable Kerberos Authentication on a SQL Server Failover Cluster (http://go.microsoft.com/fwlink/?LinkId=157417).
SQL Server Service Account	Grant permission to create SPN Entries	Required for proper Kerberos authentication. For more information, see How to use Kerberos authentication in SQL Server (http://go.microsoft.com/fwlink/?LinkId=157420).
SQL Server nodes, SQL cluster name	Create SPN entries for user SQL Server Service Account	Required for proper Kerberos authentication. For more information, see How to use Kerberos authentication in SQL Server

Affected artifact	Change	Reference and details
		(http://go.microsoft.com/fwlink/?LinkId=157420).
SQL Network Name cluster resource	DNS Registration must succeed, Enable Kerberos Authentication	Required for proper Kerberos authentication
SQL Server Surface configuration	Enable Remote Direct Administrator Connection	Required by SQL Browser Service to function properly which is required by SQL Clients (BizTalk/ASP.NET) in order to correctly locate SQL Server named instance
BizTalk Application Users Group	Grant Execute permission on sp_help_jobhistory in msdb database	Required by BizTalk Server

See Also

[Checklists for Other Important Tasks](#)

Planning the Environment for BizTalk Server

The planning section of the operations guide describes roles and responsibilities associated with a BizTalk Server environment. It includes planning recommendations for the application and data tiers of a BizTalk Server environment, and it provides planning recommendations for the release management stages of a BizTalk solution.

As the saying goes, "If you fail to plan, you plan to fail." While there are certainly exceptions to this sage advice, successful implementation of a production BizTalk solution is not one of them. This introductory topic to the planning section provides a high-level overview of the decisions you should make when planning your BizTalk solution.

Deciding Whether BizTalk Server Is the Right Tool for the Job

BizTalk Server can be thought of as a "business integration engine." At its core, BizTalk Server is designed to integrate disparate business systems, processes, and messages. For example, a business system that exchanges messages that adhere to the EDI standard may need to integrate with a business system that exchanges messages that conform to the RosettaNet standard. Or an internal business system that uses SAP may need to communicate with another internal business system that stores data in a Microsoft SQL Server database. Or perhaps a business system that can only send or receive messages using the FTP protocol needs to exchange messages with a business system that can only use the HTTP protocol.

SQL Server accommodates the integration of such disparate systems by acting as the middleman for message delivery between the systems. BizTalk Server supports a wide range of industry-standard transport protocols, document exchange formats, and line of business applications.

BizTalk Server also provides powerful business process automation capabilities in the BizTalk Orchestration engine. You use BizTalk Orchestration Designer to create visual representations of business processes which can be built into executable code that is run in the BizTalk Orchestration engine.

BizTalk Server also includes several other features that facilitate business integration including a message workflow engine, a Business Rule Engine (BRE), and technologies for information workers such as Business Activity Monitoring (BAM).

For more information about using BizTalk Server business process management functionality, see [White Paper: Microsoft and BPM—Technical Overview](http://go.microsoft.com/fwlink/?LinkId=106015) (http://go.microsoft.com/fwlink/?LinkId=106015). To know more about the different integration technologies offered by Microsoft and the advantages one has over the other, see [Understanding Microsoft Integration Technologies](http://go.microsoft.com/fwlink/?LinkId=158452) (http://go.microsoft.com/fwlink/?LinkId=158452).

Certain integration scenarios are better suited to other Microsoft products. If your primary focus is upon any of the following scenarios, consider using these Microsoft products instead of BizTalk Server:

Scenario	Product to use
User provisioning	<p>Microsoft Identity Lifecycle Manager 2010</p> <p>For more information about Microsoft Identity Lifecycle Manager 2010, see Microsoft Identity Lifecycle Manager 2010 FP1 (http://go.microsoft.com/fwlink/?LinkId=204577).</p>
Data replication between systems	<p>SQL Server Replication</p> <p>For more information about SQL Server 2008 SP1 replication, see SQL Server 2008 R2 Replication (http://go.microsoft.com/fwlink/?LinkId=69978).</p>
Data extraction, transform, and load (ETL)	<p>SQL Server Integration Services (SSIS)</p> <p>For more information about SQL Server 2008 SP1 Integration Services, see SQL Server 2008 R2 Integration Services (http://go.microsoft.com/fwlink/?LinkId=152044).</p>

Deciding Which Edition of BizTalk Server Is Right for the Job

There are four different editions of BizTalk Server 2010, each of which is targeted at specific scenarios. The four editions of BizTalk Server 2010 include:

- **Enterprise** - Designed for customers with enterprise-level requirements for high volume, reliability, and availability.
- **Standard** - Designed for businesses with moderate volume and deployment scale requirements.
- **Branch** - Specialty version of BizTalk Server designed for hub and spoke deployment scenarios including RFID.
- **Developer** - Provides all of the functionality of the Enterprise Edition for development and testing purposes and is available as the BizTalk Server 2010 Evaluation Edition at no cost for evaluation purposes. When installed as the Evaluation Edition, BizTalk Server 2010 will function for 120 days.
- **RFID Enterprise** - Designed to provide a scalable, extensible platform for development, deployment, and management of rich RFID and sensor solutions, includes BizTalk RFID Server and BizTalk RFID Mobile.

For more information about the different editions of BizTalk Server 2010, see [Microsoft BizTalk Server Editions](http://go.microsoft.com/fwlink/?LinkId=108051) (<http://go.microsoft.com/fwlink/?LinkId=108051>).

Planning for Message Load

Once you have determined that BizTalk Server meets your business integration needs, the next thing that you should determine is the message load that the BizTalk solution will be expected to process. This is an important decision because different editions of BizTalk Server have different scale-up and scale-out capabilities.

The key to determine message load is to perform load testing to determine the Maximum Sustainable Throughput (MST) and the Maximum Sustainable Tracking Throughput (MSTT) of the BizTalk solution. For more information about measuring maximum sustainable throughput, and performance best practices in general, see the [BizTalk Server 2009 Performance Guide](http://go.microsoft.com/fwlink/?LinkID=150492) (<http://go.microsoft.com/fwlink/?LinkID=150492>).

Planning for Expansion

Consider implementing a BizTalk solution using the Enterprise edition of BizTalk Server if you will be adding a significant number of trading partners, will need to use host clustering, or will need to scale out to multiple computers running BizTalk Server in the BizTalk group. The Standard and Branch editions of BizTalk Server do not accommodate multiple computers running BizTalk Server in a group or host clustering.

In This Section

- [BizTalk Server Roles and Responsibilities](#)
- [Planning the BizTalk Server Tier](#)
- [Planning the Database Tier](#)
- [Planning the Development, Testing, Staging, and Production Environments](#)

BizTalk Server Roles and Responsibilities

The development, maintenance, and administration of a BizTalk Server environment encompass several different roles with distinct responsibilities. This topic describes these roles, the responsibilities associated with the roles, and tools that can be used to complete the tasks associated with each role. This topic also provides links to Operational Readiness checklist tasks associated with each role.

Roles and Responsibilities for a BizTalk Server Environment

Role	Responsibilities	Tools	Operations Checklist Tasks
BizTalk Server Administrator	<p>Administration</p> <p>Maintain environment uptime by monitoring BizTalk Server exceptions and resolving BizTalk Server specific issues.</p> <p>Platform Modification</p> <ul style="list-style-type: none"> • Create new hosts • Add and configure adapters • Create new Message Box databases 	<p>BizTalk Server Administration console. The BizTalk Server Administration console is the primary management tool for BizTalk Server. It provides a graphical user interface for performing deployment operations for a BizTalk application. For example, you can start the Import, Installation, and Export Wizards as well as add and remove an application's artifacts and make other modifications to the application.</p> <p>Scripting and programmability APIs. Use Microsoft Windows Management Instrumentation (WMI) or the BizTalk Explorer Object Model to create and run scripts that automate administrative tasks.</p> <p>The WMI object model exposes and simplifies administrative APIs. All administration APIs expose some form of the following operations on every object they manage:</p> <ul style="list-style-type: none"> • Create • Enumerate • Modify • Delete <p>WMI exposes this functionality in a consistent manner for all WMI objects.</p> <p>Business Activity Monitoring (BAM). Business Activity Monitoring (BAM) provides an observation-based tracking framework that can be used to capture operational or business process metrics. Metrics are captured in SQL Server where they can subsequently be reported on using a reporting mechanism that is</p>	<ul style="list-style-type: none"> • Checklist: Configuring Windows Server • Checklist: Configuring Internet Information Services • Checklist: Configuring BizTalk Server • Checklist: Monitoring

Role	Responsibilities	Tools	Operations Checklist Tasks
	<ul style="list-style-type: none"> • Add additional BizTalk servers as required for scale out • Create new BizTalk groups <p>Platform Monitoring</p> <ul style="list-style-type: none"> • Monitor for BizTalk Server platform exceptions • Debug application errors. • Manage load across the BizTalk Server group. 	<p>most appropriate for the situation. BAM uses a Microsoft Office Excel workbook to provide business users with the ability to see a real-time comprehensive view of business processes.</p> <p>Health and Activity Tracking. Use the Group Hub page in BizTalk Server Administration Console to track the health of your BizTalk Server implementation, identify bottlenecks, and monitor the BizTalk Server environment. You can view the technical details of a particular orchestration, pipeline, or message instance, as well as see the message flow of a particular message that enters the system. Business users can view, monitor, and query tracked data. In addition, users can create custom views and queries, and save them for reuse in other locations. Business analysts and end users can track the state of their business processes by viewing both live and archived data.</p>	<p>Operational Readiness</p>
Release Management	<p>Deployment and Updating BizTalk Applications</p> <ul style="list-style-type: none"> • Deploy new applications into 	<p>BTSTask command-line tool. BTSTask enables you to perform many administrative tasks from the command line.</p>	<ul style="list-style-type: none"> • Checklist: Deploying an Application • Checklist:

Role	Responsibilities	Tools	Operations Checklist Tasks
	<p>the BizTalk Server environment</p> <ul style="list-style-type: none"> • Manage strong name keys <p>For more information about deploying and updating BizTalk applications, see Managing Applications</p>		<ul style="list-style-type: none"> • Exporting Bindings from One Application to Another • Checklist: Updating an Assembly • Checklist: Updating an Application Using Side-by-Side Versioning
Operations Support	<p>Support Change Management</p> <ul style="list-style-type: none"> • Support new service solutions • Support new technology 	<p>Performance Analysis of Logs (PAL) Toolhttp://go.microsoft.com/fwlink/?LinkID=98098 (http://go.microsoft.com/fwlink/?LinkID=98098). Use this tool for analyzing performance counter log files.</p> <p>Microsoft BizTalk LoadGen 2007 Toolhttp://go.microsoft.com/fwlink/?LinkID=59841 (http://go.microsoft.com/fwlink/?LinkID=59841). This is a load generation tool for simulating message transmission load on BizTalk Server. You can use it to run performance and stress tests against a BizTalk</p>	<ul style="list-style-type: none"> • Checklist: Providing High Availability with Fault Tolerance or

Role	Responsibilities	Tools	Operations Checklist Tasks
	<p>gies</p> <ul style="list-style-type: none"> • Support new systems • Support applications • Support hardware changes • Support configuration management • Support release management <p>Normal Operating Procedures Effective, efficient execution of day-to-day tasks including the following:</p> <ul style="list-style-type: none"> • Security administration • System administration • Network administration • Service monitoring 	<p>deployment for BizTalk Server versions 2004 and later.</p> <p>BizTalk Server 2006 Orchestration Profilerhttp://go.microsoft.com/fwlink/?LinkId=102209 (http://go.microsoft.com/fwlink/?LinkId=102209). This is used to view orchestration tracking data for a specified period of time; helpful for determining where performance bottlenecks exist in orchestrations.</p> <p> Note Use of this tool is not supported by Microsoft, and Microsoft makes no guarantees about the suitability of this programs. Use of this program is entirely at your own risk.</p> <p> Note This tool can be used with BizTalk Server 2010 as well.</p> <p>Tools and Utilities to use for Troubleshootinghttp://go.microsoft.com/fwlink/?LinkID=154416 (http://go.microsoft.com/fwlink/?LinkID=154416).</p>	<ul style="list-style-type: none"> • Load Balancing • Checklist: Increasing Availability with Disaster Recovery • Performing Bottleneck Testing and Tuning • Performing Load and Throughput Testing • Performing Availability Testing

Role	Responsibilities	Tools	Operations Checklist Tasks
	<p>g and control</p> <ul style="list-style-type: none"> • Directory services administration • Storage management • Job scheduling <p>Support Procedures Timely resolution of problems and inquiries.</p> <ul style="list-style-type: none"> • Service desk support • Incident management • Problem management <p>Environment Optimization</p> <ul style="list-style-type: none"> • Cost management • Performance optimization 		

Role	Responsibilities	Tools	Operations Checklist Tasks
	<ul style="list-style-type: none"> • Service level management • Capacity management • Availability management • Work force management 		
Developer	<p>Application Monitoring</p> <ul style="list-style-type: none"> • Support application exceptions that are not resolved by the BizTalk Server administrator. • Provide updates to source code for resolving issues. <p>New and Existing Application</p>	<p>Visual Studio 2010 Development platform used to create, modify, and debug BizTalk Server solution source code.</p> <p>Testing the Applicationhttp://go.microsoft.com/fwlink/?LinkId=159595 (http://go.microsoft.com/fwlink/?LinkId=152653). Useful for unit testing .NET code.</p> <p>DebugView for Windowshttp://go.microsoft.com/fwlink/?LinkId=102210 (http://go.microsoft.com/fwlink/?LinkId=102210). Tool for monitoring kernel-mode and Win32 debug output either locally or remotely.</p>	<p>Performing Unit Testing</p>

Role	Responsibilities	Tools	Operations Checklist Tasks
	<p>Development</p> <p>Create new applications and update existing application functionality on the BizTalk Server platform.</p>		
Database Administrator	<p>Database Management</p> <ul style="list-style-type: none"> • Define appropriate database space allocation • Monitor database usage • Tune SQL Server • Set up backup and restore jobs for all SQL Server databases. • Ensure that maintenance 	<p>SQL Server Profiler SQL Server Profiler can be used to capture Transact-SQL statements that are sent to SQL Server and the SQL Server result sets from these statements. Since BizTalk Server is tightly integrated with SQL Server, the analysis of a SQL Server Profile trace can be a useful tool for analyzing problems that may occur in BizTalk Server when reading from and writing to SQL Server databases. For information about how to use SQL Server Profiler, see Introducing SQL Server Profiler (http://go.microsoft.com/fwlink/?LinkId=152656).</p> <p>SQL Server Management Studio SQL Server Management Studio can be used to execute SQL statements directly against SQL Server databases. This functionality may be useful for querying the BizTalk Server databases or for updating the BizTalk Server databases in certain scenarios. For more information about SQL Server Management Studio, see Using SQL Server Management Studio (http://go.microsoft.com/fwlink/?LinkId=152657).</p>	<p>Checklist: Configuring SQL Server</p>

Role	Responsibilities	Tools	Operations Checklist Tasks
	<p>scripts are run regularly for the Tracking and Orchestration Persistence database s.</p> <ul style="list-style-type: none"> • Work with the trend analysis specialist to identify appropriate thresholds. • Complete standard SQL Server maintenance procedures. 		

See Also

[Tools for Testing](#)

[Operational Readiness Checklists](#)

Planning the BizTalk Server Tier

This section contains topics with planning recommendations for the application tier of a BizTalk solution.

In This Section

- [Planning for Sending and Receiving](#)
- [Planning the BizTalk Solution](#)
- [Planning for Single Sign-On](#)
- [Planning for Tracking](#)
- [Planning for Publishing Web Services](#)
- [Planning for Consuming Web Services](#)
- [Planning for Performance](#)
- [Planning for High Availability and Disaster Recovery](#)
- [Planning for Testing](#)
- [BizTalk Server 64-Bit Support](#)

Planning for Sending and Receiving

Nearly every document that is processed by BizTalk Server is received by a BizTalk Server receive adapter, and sent from BizTalk Server using a BizTalk Server send adapter. Because BizTalk Server adapters figure so prominently in any BizTalk Server environment, it is important to plan ahead to determine which adapters or accelerators you will be using and how to correctly configure these adapters and/or accelerators.

Determining Which Adapters and Accelerators You Will Use

Confer with your trading partners in advance to determine which adapters and accelerators you will need to facilitate sending and receiving documents between your organization and your trading partners, and between your BizTalk applications and your internal business applications. Design your BizTalk Server architecture to be flexible enough to accommodate adding additional adapters or accelerators in the event that you establish relationships with additional trading partners in the future.

Functionality Supported by BizTalk Adapters

The table in this section lists the primary benefit of each native adapter and whether the adapter provides the following features:

- **Transaction support** The ability to send and receive documents under the context of a distributed transaction coordinator (DTC) transaction. This functionality is required for maintaining ordered message delivery and to guarantee that documents are not duplicated or lost.



Note

If you encounter problems with MSDTC, review the topic [Troubleshooting Problems with MSDTC](http://go.microsoft.com/fwlink/?LinkID=154693) (http://go.microsoft.com/fwlink/?LinkID=154693).

- **Two-way communication support (Request/Response or Solicit/Response)** The ability to send a document and process a response message from the destination or to receive a document and send a response message to the source.
- **In-order receive support.** The ability to publish received documents to the MessageBox database in the exact order that the documents were received.

 **Note**

Certain adapters can enforce ordered document delivery at the receive location level, while other adapters cannot. Ordered delivery can still be enforced at the send port level for those adapters which do not support ordered document delivery at the receive location level but doing so may incur a performance penalty. For more information about ordered delivery of messages, see the topic [Ordered Delivery of Messages](http://go.microsoft.com/fwlink/?LinkID=155751) (http://go.microsoft.com/fwlink/?LinkID=155751).

- **SSO enabled.** The ability to use SSO authentication when sending or receiving documents with the adapter.

Adapter	Primary Benefit	Transaction Support	Two-Way Communication Support	In-Order Receive Support	SSO Enabled
File	Easy to use	No	No	No	No
FTP	Is widely used for business-to-business communications	No	No	No	Yes
HTTP(s)	Is widely used for business-to-business communications	No	Request/Response and Solicit/Response	No	Yes
SOAP	Supports the use of Web services	No	Request/Response and Solicit/Response	No	Yes
MSMQ	Supports guaranteed once-only delivery of messages between BizTalk Server and Microsoft	Yes	No	Yes	No

Adapter	Primary Benefit	Transaction Support	Two-Way Communication Support	In-Order Receive Support	SSO Enabled
	Message Queuing				
MQ Series	Supports guaranteed once-only delivery of messages between BizTalk Server and IBM WebSphere MQ for Windows platforms	Yes	No	Yes	Yes
SQL	Supports direct communication between BizTalk Server and SQL Server databases	Yes	Solicit/Response only	No	No
Windows SharePoint Services	Enables the exchange of XML and binary messages between BizTalk Server and SharePoint document libraries	No	No	No	No
POP3	Supports receiving documents through e-mail	No	No	No	No
SMTP	Supports sending documents through e-mail	No	No	No	No
EDI	Supports processing of	No	No	No	No

Adapter	Primary Benefit	Transaction Support	Two-Way Communication Support	In-Order Receive Support	SSO Enabled
	business documents that conform to the EDI standard				
Custom	Supports your legacy system	Yes, requires custom code.	Yes, requires custom code.	Yes, requires custom code.	Yes, requires custom code.
WCF-WSHttp	Supports WS-* standards over the HTTP transport	Yes, transactions are supported on WsHTTP (only WS-Transactions)	Request/Response and Solicit/Response	No	Yes
WCF-BasicHttp	Communicates with ASMX-based Web services and clients and other services that conform to the WS-I Basic Profile 1.1 using HTTP or HTTPS	No	Request/Response and Solicit/Response	No	Yes
WCF-NetTcp	Supports WS-* standards over the TCP transport	Yes	Request/Response and Solicit/Response	No	Yes
WCF-NetMsmq	Supports queuing by leveraging Microsoft Message Queuing (MSMQ) as a transport	Yes	No	Yes	Yes

Adapter	Primary Benefit	Transaction Support	Two-Way Communication Support	In-Order Receive Support	SSO Enabled
WCF-NetNamedPipe	Provides a fast transport for cross-process communication on the same computer (only for WCF apps)	Yes	Request/Response and Solicit/Response	No	Yes
WCF-Custom	Enables the use of WCF extensibility features	Yes.	Yes.	Yes, as long as the binding supports it.	Yes.
WCF-CustomIsolated	Enables the use of WCF extensibility features over the HTTP transport	Yes.	Yes.	No.	Yes.

Line of Business Adapters

Following is a list of the Line of Business (LOB) adapters provided by Microsoft.



Note

With the exception of the Microsoft BizTalk Adapter v2.0 for mySAP™ Business Suite (the Adapter), none of the Line of Business adapters are supported on Windows Vista.

Adapter	Description	Supported Versions
SAP (also referred to as "the Adapter")	Enables exchange of Intermediate Document (IDOC), BAPI, and remote function call (RFC) messages between BizTalk Server and an SAP R/3® system.	SAP R/3 4.x and R/3 6.20 (Enterprise)
PeopleSoft Enterprise	Enables exchange of Component Interface (CI) messages between BizTalk Server and a PeopleSoft	PeopleTools Versions 8.17.02, 8.43, 8.45, 8.46 and 8.48

Adapter	Description	Supported Versions
	system.	
JD Edwards OneWorld XE	Enables exchange of Business Function messages between BizTalk Server and a JD Edwards OneWorld system.	B7.3.3.3 with SP23 and JDE 8.0 (B7.3.3.4)
JD Edwards EnterpriseOne	Enables exchange of Business Function messages between BizTalk Server and a JD Edwards EnterpriseOne system.	8.10 & 8.11 with Tools Release 8.93, 8.94, 8.95 and 8.96
ODBC Adapter for Oracle Database	Enables reading and writing information from and to an Oracle Server database.	Oracle 8i (8.1.6.0), 9i (9.2.0.1), or 10g
Siebel eBusiness Applications	Enables exchange of Business Components and Business Service messages between BizTalk Server and a Siebel eBusiness Application.	7.0, 7.5.*, 7.7.*, and 7.8.*
TIBCO Rendezvous	Enables exchange of XML and binary data format messages between BizTalk Server and TIBCO Rendezvous.	7.3
TIBCO Enterprise Message Service	Enables exchange of XML and binary data format messages between BizTalk Server and a TIBCO EMS server providing a tightly integrated and reliable application infrastructure.	4.2
WebSphere MQ	Enables exchange of messages between BizTalk Server and IBM WebSphere MQ.	5.3 with Fix Pack 10 or later and 6.0 with Fix Pack 1.1 or later

For more information about the LOB adapters available with BizTalk Server 2010, see [Adapters included with BizTalk Server 2010](http://go.microsoft.com/fwlink/?LinkId=152664) (http://go.microsoft.com/fwlink/?LinkId=152664).

BizTalk Adapter Pack

Microsoft BizTalk Adapter Pack contains WCF-based adapters to provide connectivity to LOB applications such as Oracle Database, Oracle E-Business Suite, SAP, Siebel, and SQL Server. For a list of adapters available with BizTalk Adapter Pack, see [BizTalk Adapter Pack](http://go.microsoft.com/fwlink/?LinkId=152665) (<http://go.microsoft.com/fwlink/?LinkId=152665>).

Important

You can use the BizTalk Adapter Pack Migration Tool to migrate BizTalk projects for LOB adapters to BizTalk projects for WCF-based LOB adapters available with the BizTalk Adapter Pack. You can download the BizTalk Adapter Pack Migration Tool from [BizTalk Adapter Pack Migration Tool](http://go.microsoft.com/fwlink/?LinkId=153328) (<http://go.microsoft.com/fwlink/?LinkId=153328>). To know more about migrating LOB adapters to WCF-based LOB adapters included with the BizTalk Adapter Pack, see the [Microsoft BizTalk Adapter 2.0 Migration Whitepaper](http://go.microsoft.com/fwlink/?LinkId=158848) (<http://go.microsoft.com/fwlink/?LinkId=158848>).

BizTalk Accelerators

While BizTalk adapters accommodate sending and receiving documents with a particular protocol, BizTalk accelerators are designed to accommodate the exchange of documents in accordance with a particular industry standard. For a list of the available BizTalk accelerators, see [Microsoft BizTalk Server Accelerators](http://go.microsoft.com/fwlink/?LinkId=103609) (<http://go.microsoft.com/fwlink/?LinkId=103609>).

Configuring Your Domain When Exposing Transports to the Internet

In order to facilitate the sending and receiving of documents between your organization and external trading partners, it may be necessary to expose transports on a public facing site that is accessible from the Internet. Under these circumstances, the following domain configuration is recommended:

- **Employ a perimeter network domain, (also known as a demilitarized zone (DMZ) or screened subnet), to house servers to provide Internet related services for your organization**

The perimeter network domain should contain servers which house the physical locations where Internet-facing transports route documents between computers running BizTalk Server and your trading partners. The perimeter network firewall should only open the ports required to allow communications to and from the Internet facing transports. There should be no computers running BizTalk Server, BizTalk Server receive locations, or Enterprise Single Sign-On server computers in the perimeter network domain. Documents sent/received to/from transports in the perimeter network domain should be routed from the Internet-facing firewall to the firewall protecting the processing domain using Internet Security and Acceleration Server (ISA) Server Web Publishing and Server Publishing. For more information about using ISA Server Web and server publishing, see [Publishing Concepts in ISA Server 2006](http://go.microsoft.com/fwlink/?LinkId=86359) (<http://go.microsoft.com/fwlink/?LinkId=86359>).

Note

As an added measure of security, consider using public key infrastructure (PKI) digital certificates for purposes of document encryption and decryption, document

signing and verification (non-repudiation) for documents sent to or received from trading partners via Internet facing transports in this domain.

The following transports are typically used on the perimeter network domain that is accessible from the Internet:

- FTP - to receive documents using the FTP protocol
- SMTP - to send documents using the SMTP protocol
- HTTP - to receive documents using the HTTP protocol
- SOAP - to receive documents using SOAP
- POP3 - to receive document using the POP3 protocol
- **Employ a processing domain to house servers that contain the BizTalk Server receive and send handlers and a BAM portal server**

Document flow between the external facing transports in the perimeter domain and BizTalk adapters in the processing domain should be routed through a firewall between these domains. The processing domain should house the BizTalk Server ports, receive locations, pipelines, maps, schemas, and assemblies used to receive, route, and send messages. The processing domain should also contain servers for the BAM portal. The number of computers running BizTalk Server in the processing domain will depend on the number of hosts and host instances required to meet the performance needs of your organization.

Important

Ensure that you create sufficient isolated host instances in the processing domain to accommodate traffic flowing between the HTTP and SOAP transports in the perimeter domain and the HTTP and SOAP adapters in the processing domain. If a large portion of the document traffic between your organization and your trading partners flows through the HTTP and SOAP transports, there must be sufficient processing bandwidth in the processing domain to handle the document flow.

- **Employ additional domains to provide further layers of isolation and security for your environment**

These domains should include:

- A services domain which is trusted by the processing domain and which is needed to process messages successfully. Servers in the services domain typically run BizTalk orchestrations, pipelines, Enterprise Single Sign-On (SSO) services, the Business Rule Engine, and may include other business processes.
- A data domain for the computers running SQL Server used by BizTalk Server.
- A corporate domain for servers and desktop computers to provide services to information workers in your organization.

For more information about the domain topologies recommended for various BizTalk Server architectures, see [Sample BizTalk Server Architectures](http://go.microsoft.com/fwlink/?LinkId=155750) (<http://go.microsoft.com/fwlink/?LinkId=155750>).

High Availability Considerations

High availability can be provided for most adapters by running adapter handler host instances on multiple BizTalk servers in a BizTalk group. That way, if one adapter handler host instance fails, another adapter handler host instance is available to continue processing. There are, however, exceptions to doing this. In some cases running multiple adapter handler host instances can cause problems with contention. For example contention problems can occur when running multiple instances of the POP3 and FTP adapters. In these circumstances, high availability can be provided for the adapter by running the adapter handler host instance in a clustered BizTalk host.

For more information about providing high availability for an adapter handler host instance through host clustering, see [Considerations for Running Adapter Handlers within a Clustered Host](http://go.microsoft.com/fwlink/?LinkID=151284) (<http://go.microsoft.com/fwlink/?LinkID=151284>). For more information about providing high availability for BizTalk hosts, see [High Availability for BizTalk Hosts](#).

Performance Considerations

SOAP Adapter Performance Considerations

For information about optimizing the performance of the SOAP adapter, see [Configuration Parameters that Affect Adapter Performance](http://go.microsoft.com/fwlink/?LinkID=154200) (<http://go.microsoft.com/fwlink/?LinkID=154200>).

MQSeries Adapter Performance Considerations

Disable transaction support and ordered delivery if not required for MQSeries adapter receive locations When an MQSeries adapter receive location is configured with the **Transaction Supported** option set to **Yes**, or the **Ordered** option set to **Order with Stop**, then each message picked up by the receive location will be processed in the context of an Microsoft Distributed Transaction Coordinator (MSDTC) transaction. Because there is additional overhead incurred when processing messages in the context of an MSDTC transaction, these options should not be enabled if ordered delivery or transaction support is not required for the MQSeries adapter receive location.

Planning for Ordered Message Delivery

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. The following considerations apply when implementing ordered delivery of messages:

Configuring Ordered Message Delivery

Ordered message delivery can be configured in the following places:

- Receive shape in an orchestration
- Receive location for certain adapters
- 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 MessageBox database before delivering the next message, in another batch, to the adapter.

Ordered Delivery for Custom Adapters

For a custom receive adapter to preserve the order of messages when submitting them to BizTalk Server, the adapter must be developed with the following functionality:

- After submitting a batch of messages, your custom receive adapter should wait for the BatchComplete call back from BizTalk Server before submitting the next batch. For more details, see [Interfaces for a Batch-Supported Receive Adapter](http://go.microsoft.com/fwlink/?LinkId=155752) (<http://go.microsoft.com/fwlink/?LinkId=155752>).
- If a message fails in the pipeline, it should be suspended, preferably as non-resumable. Use the BTS.SuspendAsNonResumable message context property in BizTalk Server to flag the message appropriately.



Note

Message order can be broken if a suspended message is later resumed. If you do not want this behavior, suspend failed messages as non-resumable.

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, examples of such adapters are MSMQ and MQSeries. 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**.

See Also

[Planning the Environment for BizTalk Server](#)

Planning the BizTalk Solution

One of the primary design goals of BizTalk Server is to provide maximum flexibility for accommodating as many processing scenarios as possible. Because of this great flexibility, one of the primary challenges facing developers of a BizTalk solution is to determine how to make best use of the features available in BizTalk Server to best meet their business needs. Planning the BizTalk Server can be broken down into distinct phases which are summarized below.

Scoping the Solution

Performance Considerations

Consider the following when scoping your BizTalk solution:

- Which adapters and/or accelerators are required?
- What are the requirements for implementing orchestrations in the solution?
- Document throughput requirements: What are the maximum sustainable throughput requirements for the solution?
- Latency requirements: How responsive does the solution need to be for solicit-response and request-response scenarios?
- How well does the solution recover from periods of peak document load?
- What are the high availability requirements of the solution?
- What are the document tracking requirements of the solution?
- What are the performance characteristics of any dependent applications such as a remote Web service or other system? If dependent applications do not keep up with the required load then the overall system performance will be degraded accordingly.
- Would the BizTalk application be consuming databases not related to BizTalk Server? For example, if the BizTalk application is consuming tables in a SQL Server database using the SQL adapter, are the tables efficiently configured?

Hardware Considerations

When scoping the solution, create a high-level hardware diagram that includes the following:

- Computer architecture (such as x86, x64, and IA64)
- CPU requirements (such as type, speed, number, cores, and use of hyperthreading)
- RAM requirements for each computer
- Local disk storage (type, size, speed)
- SAN (storage requirements: number of LUNS; SAN card type)
- Network cards (number in each computer, 100 megabits (Mbps) versus 1 Gigabit (1 Gbps).)
- How will firewalls be deployed in the solution?
- Will Network Load Balancing hardware be used?
- Are specific computers to be clustered?
- Would you be using a virtual environment involving Microsoft Hyper-V Server or any other virtualization products?

Planning the Solution

Solution Milestones Timeline

Create a schedule with milestones for completing specific aspects of your BizTalk solution. Setting specific milestones will increase the likelihood that the solution will be completed in a timely manner.

Non-Microsoft Software Considerations

Consider the following when non-Microsoft software will be used with the solution:

- Determine how the software or hardware required be obtained.

- Plan for capacity and sizing to ensure that non-Microsoft software does not become a bottleneck in your solution.
- Determine a plan of action for installing required non-Microsoft software.
- Create a plan of action for configuring and optimizing required non-Microsoft software.

Preparing for the Solution

Include the following elements in your preparation phase:

Detailed Design of the Solution Platform

A detailed solution design facilitates communications and avoids assumptions, which will improve the agility and effectiveness of all activities. You should fully document the following elements:

- BizTalk Server databases and how they will be distributed across computers.
- BizTalk Host design and descriptions of each host and their instances.
- Description of each orchestration.
- Description of each pipeline.
- Description of custom components such as .NET assemblies and COM+ components.

Message Flow Diagrams

Create detailed message flow diagrams to help avoid any confusion or false assumptions regarding what is supposed to be happening to messages during processing. The following details should be considered when creating the message flow diagrams:

- Describe the lifecycle of each type of message from the time it arrives at a receive location until all resulting messages are sent and all related processing is completed.
- Describe how processing changes for error conditions.
- Include details about correlation, delivery notifications, and acknowledgements.
- Include performance requirement information regarding latency and throughput.

Non-Microsoft Software Details

All non-Microsoft software that is used should be fully documented as part of the detailed solution design.

Detailed Hardware Stack

Building on the previously created high level hardware diagram, the following hardware information should be fully documented:

- Processors
 - Type
 - Speed
 - Number of cores
 - Hyperthreading
- Memory
 - Amount
 - Speed
 - Parity

- Network
 - Number of network interface cards (NICs)
 - Speed of network
- SAN
 - Number of SAN cards in each computer
 - Number of logical unit numbers (LUNs) for each computer and purpose for each LUN
 - Speed of storage area network (SAN) Cards
 - SAN card configuration details
 - SAN disk allocation, formatting, and partitioning
- Disk
 - Local disk details for each computer
 - Formatting used for local disks
 - Partitioning details for local disks
- Cache
 - L2 Cache amount
 - L3 Cache amount

Detailed Software Stack

The following software information should be documented:

- Specific operating system versions, editions, and architecture
- Specific operating system features
- Specific software installed on each computer
- Specific drivers
- Service Packs and other updates
- Configuration values for all software and operating system features used if they vary from default values

Building Out the Environment for the Solution

Detailed instructions for installing BizTalk Server and the required dependency software can be found in the BizTalk Server Installation Guides available for download at [BizTalk Server 2010 Documentation](http://go.microsoft.com/fwlink/?LinkId=183138) (http://go.microsoft.com/fwlink/?LinkId=183138).

See Also

[Planning the BizTalk Server Tier](#)

Planning for Single Sign-On

Enterprise Single Sign-On (SSO) is a critical component of the BizTalk Server environment. The BizTalk Server run time cannot function without the SSO service because all BizTalk Server adapter configuration information is encrypted and stored in the SSO database and BizTalk Server accesses this information via the SSO service. This adapter configuration information is

not accessible if the SSO service is not running on the BizTalk server or if the SSO service does not have access to the SSO master secret server.

Your SSO implementation should include the following plans.

Backing Up the Master Secret

If the SSO master secret server fails and you lose the key, or if the key becomes corrupted, you will not be able to retrieve data stored in the SSO database. You must back up the master secret, or you risk losing data from the SSO database. Therefore it is absolutely critical that the SSO master secret is backed up and stored in a secure location. For information about backing up the SSO master secret, see [How to Back Up the Master Secret](http://go.microsoft.com/fwlink/?LinkId=151395) (<http://go.microsoft.com/fwlink/?LinkId=151395>).

Configuring SSO for High Availability

Because SSO is such a critical component of a BizTalk Server environment, the SSO master secret server should be configured for high availability. If at all possible, the Enterprise Single Sign-On service on the master secret server should be clustered by following the steps in [Clustering the Master Secret Server](#). In the event that you do not have access to a Windows Server cluster, you can still provide high availability for the Enterprise Single Sign-On service on the master secret server by following the steps documented in the topic [Designating a New Master Secret Server Manually](#).

Following Established SSO Security Recommendations

Follow SSO security recommendations described in the topic [SSO Security Recommendations](#) (<http://go.microsoft.com/fwlink/?LinkId=155753>).

Planning for Tracking

Message tracking is the process by which parts of a message instance, such as the message body, message properties, and metadata are stored in a database, typically for archival purposes. Message instance parts that are tracked can subsequently be viewed by running queries from the Group Hub page in the BizTalk Server Administration console. In addition to accessing archived data, you can also view live data, which can be a helpful tool for identifying and fixing problems in a development or staging environment.

Since the process of message tracking can be very resource intensive, you should review this topic before creating your plan.

For more information about tracking, see [Health and Activity Tracking](#) (<http://go.microsoft.com/fwlink/?LinkId=154187>).

Configuring and Enabling the DTA Purge and Archive SQL Agent Job

This job archives and purges old data from the BizTalk Tracking database, thus keeping it from becoming too large. This is essential for a healthy BizTalk Server system. A large tracking database will begin to affect the performance of the tracking host and any other processes that query the tracking database.

- **Ensure that the DTA Purge and Archive SQL Agent job is properly configured, enabled, and successfully completing.** This job is not enabled by default because you must first configure it to include a directory where the archive files can be written.
- **Ensure that the job is able to purge the tracking data as fast as the incoming tracking data is generated.** It is acceptable for the job to get behind during peak load times, but it should always be able to catch up. If the purge job gets behind and is never able to catch up, the BizTalk Tracking database will continue to grow, and performance will eventually be adversely affected.
- **Review the soft purge and hard purge parameters to ensure you are keeping data long enough but not too long.** For more information about these parameters see [Archiving and Purging the BizTalk Tracking Database](http://go.microsoft.com/fwlink/?LinkID=153816) (http://go.microsoft.com/fwlink/?LinkID=153816).
- **If you only need to purge the old data and do not need to archive it first, then change the SQL Agent job to call the stored procedure “dtasp_PurgeTrackingDatabase”.** This skips the archive step, and just does the purge. For more information about this stored procedure and changing the SQL Agent job to use it, see [How to Purge Data from the BizTalk Tracking Database](http://go.microsoft.com/fwlink/?LinkID=153817) (http://go.microsoft.com/fwlink/?LinkID=153817).
- **If you need to keep the BizTalk Tracking database archive files, ensure that you have a process in place to successfully restore and use them.**
- **If you are having performance issues that are momentarily addressed by purging the BizTalk tracking database, and you want to configure BizTalk to no longer collect tracking information, you may want to consider turning off global tracking.** For information about how to turn off global tracking, see the topic [How to Turn Off Global Tracking](http://go.microsoft.com/fwlink/?LinkID=154193) (http://go.microsoft.com/fwlink/?LinkID=154193).

Creating a Dedicated Tracking Host

When the option to **Allow Host Tracking** is enabled for a host in the BizTalk Server Administration console, instances of that host will run the Tracking Data Decode Service (TDDS) to move tracked data from the BizTalk Server MessageBox database to the BizTalk Tracking database. Since TDDS may be resource intensive, consider creating a "dedicated" tracking host for which the **Allow Host Tracking** option is enabled and which does not run any other BizTalk Server processes (such as adapters or orchestrations). If your BizTalk group contains more than one BizTalk server, it is also considered a best practice to create an instance of this host on each server in the group to provide high availability for TDDS.

Testing to Measure Maximum Sustainable Tracking Throughput

Extensive message tracking is a very resource intensive activity and if not properly managed can have an extremely adverse effect on the performance of the BizTalk Server environment. Therefore, you should measure maximum sustainable tracking throughput for your BizTalk Server environment to ensure that the system is sustainable and will run indefinitely at a given message flow rate. For more information about measuring maximum sustainable tracking throughput, see [Measuring Maximum Sustainable Tracking Throughput](http://go.microsoft.com/fwlink/?LinkID=153815) (http://go.microsoft.com/fwlink/?LinkID=153815).

Best Practices for Tracking

- **Determine the information you need to track during planning** : You should decide during the planning stages which information you need to track, so that after you deploy the project you can set the tracking options and limit the amount of tracked data to give you only the information you need.
- **Do not track all messages**: We recommend that you not track all messages, because each time a message is touched, BizTalk Server makes another copy. You can instead narrow the scope by tracking only a specific port. This helps to maximize the performance of your system and to keep the databases uncluttered.
- **Set tracking on send ports and receive ports instead of on a pipeline**: If you set tracking options on pipelines, you will also set the tracking options globally for every port that uses the pipeline. This in turn may result in far more data being tracked than you intend, which will slow system performance. Instead, you can set tracking options on send ports and receive ports.
- **Take into account various factors when you size the BizTalk Tracking database**:
 - When sizing the BizTalk Tracking database, account for SQL Server factors, such as index size, by adding a contingency multiplier to your calculations.
 - When determining the size of messages in the BizTalk Tracking database, add the average size of the message context to the message size if it is significant compared to the message size.
 - To limit the size of messages in the BizTalk Tracking database, limit the number of properties that you promote. You should only use promoted properties if you need them for routing purposes, otherwise use distinguished fields.
 - If the orchestration **Shape start and end** option is enabled, take into account that a start and stop event for each shape in each orchestration instance is saved in the BizTalk Tracking database.

Planning for Publishing Web Services

BizTalk Server provides built-in support for Web services. It enables you to reuse and aggregate your existing Web services within your orchestrations.

You can also publish (expose) your orchestrations as Web services to separate the Web service logic from the business process logic, which allows you to update or replace the business logic as needed without touching the code used for the Web service logic. This functionality is referred to as implementing "modular code." In general it is considered a best practice to implement modular code where possible. Publishing Web services requires that you enable Web services and that you publish an orchestration or schema as a Web service using the BizTalk Web Services Publishing Wizard.

BizTalk Server implements support for native adapters in Web services through the use of the SOAP adapter. Native adapter support provides scalability, fault tolerance, and tracking capabilities for Web services without writing a single line of code. For more information about the SOAP adapter, see [SOAP Adapter](http://go.microsoft.com/fwlink/?LinkId=155754) (http://go.microsoft.com/fwlink/?LinkId=155754).

BizTalk Server 2010 provides support for publishing BizTalk Applications as WCF Services with Windows Azure AppFabric Service Bus endpoints. For more information see [Using AppFabric Connect for Services](#).

Planning for Web services can be divided into two categories, planning for publishing Web services and planning for consuming Web services. This topic describes the steps that you should follow for publishing Web services.

Enabling Web Services

To publish Web services, you must configure Internet Information Services (IIS), BizTalk Isolated Hosts, and Windows user and group accounts. This section provides an overview about enabling web services. For more information about enabling Web services, see the IIS documentation.

Internet Information Services 7.0

You can publish Web services to Windows systems that have IIS 7.0 or higher configured with ASP.NET 2.0. For IIS 7.0, all Web services run within the ASP.NET worker process.

IIS 7.0 uses IIS application pools for processing Web service requests. IIS 7.0 supports multiple application pools and each application pool process can run under a different user context.

BizTalk Isolated Hosts

To enable Web services, you must create at least one isolated host in BizTalk Server. Isolated hosts represent external processes, such as ISAPI extensions and ASP.NET processes that BizTalk Server does not create or control. These types of external processes must host certain adapters, such as HTTP/S and SOAP.

The BizTalk Server Configuration Manager creates the BizTalkServerIsolatedHost that BizTalk Server uses as the default isolated host. The BizTalk Isolated Host Users group is the name of the Windows group associated with this host by default. For more information about hosts and host instances, see [Managing BizTalk Hosts and Host Instances](#) (<http://go.microsoft.com/fwlink/?LinkID=154191>).

An isolated host instance can run only one adapter. If you configure the receive handlers of HTTP and SOAP adapters with the one isolated host, you must create two application pools, one application pool for each adapter.

For example, if you plan to configure two isolated hosts, as follows:

Isolated Host Name	Receive Locations
Isolated Host 1	HTTP_ReceiveLocation1A HTTP_ReceiveLocation1B SOAP_ReceiveLocation1  Note The Isolated Host 1 is used for receive handlers of both SOAP and HTTP adapters.

Isolated Host Name	Receive Locations
Isolated Host 2	HTTP_ReceiveLocation2

You may create four virtual directories, one for each receive location, as follows:

Receive Location	Virtual Directory
HTTP_ReceiveLocation1A	IIS_Virtual_Directory1A
HTTP_ReceiveLocation1B	IIS_Virtual_Directory1B
SOAP_ReceiveLocation1	IIS_Virtual_Directory1C
HTTP_ReceiveLocation2	IIS_Virtual_Directory2

Then you must create at least three application pools for the virtual directories as follows:



Note

You must create at least one application pool for each isolated host.

Virtual Directories	Application Pool	Description
IIS_Virtual_Directory1A IIS_Virtual_Directory1B	AppPool_Host1_HTTP	A separate application pool is not required because all of the receive locations have the same isolated host (Isolated Host 1), and the same protocol.
IIS_Virtual_Directory1C	AppPool_Host1_SOAP	A separate application pool is required because the receive location uses the different protocol (SOAP) from the other receive locations in the same host (Isolated Host 1).
IIS_Virtual_Directory2	AppPool_Host2_HTTP	A separate application pool is required because the receive location runs under the different host from Isolated Host 1.

Keep the following important points in mind as you create your application pools:

- You must add the user account for the application pools to the appropriate local or domain groups of the isolated hosts. For more information, see [Windows Group and User Accounts in BizTalk Server](http://go.microsoft.com/fwlink/?LinkId=155755) (http://go.microsoft.com/fwlink/?LinkId=155755).
- You need to match the user account between an isolated host instance and the corresponding application pool according to the previous tables. For more information about the relationship between user accounts of isolated host instance and application pools, see [How to Change Service Accounts and Passwords](http://go.microsoft.com/fwlink/?LinkId=155756) (http://go.microsoft.com/fwlink/?LinkId=155756).
- IIS 5.0 and IIS 5.1 in Windows 2000 Server or Windows XP do not have application pools. If you are using IIS 5.0 and 5.1, set the isolation level for the IIS virtual directories to **High**. This will create a separate COM+ application for each virtual directory. Each COM+ application will run in its own dllhost.exe.

Database Access for Single Server Installations

If BizTalk Server and the BizTalk Management database reside on the same server, you should set the user context of the ASP.NET worker process or the IIS Application Pool to the local ASPNET user account, or to a local or domain user account that has minimal privileges.

Database Access for Multiple Server Installations

If BizTalk Server and the BizTalk Management database reside on different servers, you should change the user context of the ASP.NET worker process or the IIS Application Pool to a domain user account.

When implementing a multi-server deployment, the Isolated Host Windows groups must exist on the domain that the BizTalk database servers belong to.

Minimizing Account Privileges and User Rights

You use isolated hosts to give adapters that run in external processes access to the minimal amount of required resources to interact with BizTalk Server. For a more secure deployment, you should give the user context for the external processes minimal privileges.

Security Recommendations for BizTalk Web Services Publishing Wizard

The virtual directory created by the BizTalk Web Services Publishing Wizard will inherit the access control lists (ACL) and authentication requirements from the parent virtual directory or Web site. If the parent virtual directory or Web site allows anonymous access, the BizTalk Web Services Publishing Wizard will remove that capability when creating the virtual directory.

Enabling ASP.NET 4.0 for Published Web Services

To enable ASP.NET 4.0 for published Web services, see [How to Enable ASP.NET 4.0 for Published Web Services](http://go.microsoft.com/fwlink/?LinkId=155758) (http://go.microsoft.com/fwlink/?LinkId=155758).

Using the BizTalk Web Services Publishing Wizard

For more information about publishing an orchestration as a Web service, see [Publishing an Orchestration as a Web Service](http://go.microsoft.com/fwlink/?LinkId=155761) (http://go.microsoft.com/fwlink/?LinkId=155761).

For more information about publishing schemas as a Web service, see [Publishing Schemas as a Web Service](http://go.microsoft.com/fwlink/?LinkId=155762) (http://go.microsoft.com/fwlink/?LinkId=155762).

Using AppFabric Connect for Services

For more information about publishing BizTalk Applications as WCF Services with Windows Azure AppFabric Service Bus endpoints see [Exposing BizTalk Applications on the Cloud using AppFabric Connect for Services](http://go.microsoft.com/fwlink/?LinkID=204700) (http://go.microsoft.com/fwlink/?LinkID=204700).

Planning for Publishing WCF Services

BizTalk Server introduces built-in support for Windows Communication Foundation (WCF). BizTalk Server enables you to reuse and aggregate all your existing WCF services within your orchestrations. BizTalk Server also implements support for native adapters in WCF services. Native adapter support provides scalability, fault tolerance, and tracking capabilities for WCF services without requiring you to write code. For information about the WCF adapters, see [WCF Adapters](http://go.microsoft.com/fwlink/?LinkId=155763) (http://go.microsoft.com/fwlink/?LinkId=155763).

For more information about planning for WCF Services in BizTalk Server, see [Publishing WCF Services](http://go.microsoft.com/fwlink/?LinkId=155764) (http://go.microsoft.com/fwlink/?LinkId=155764).

See Also

[Planning for Consuming Web Services](#)

Planning for Consuming Web Services

Planning for Web services can be divided into two categories, planning for publishing Web services and planning for consuming Web services. This topic describes the considerations for consuming Web services. For information about publishing Web services, see [Planning for Publishing Web Services](#).

Keep the following in mind as you create your plan:

- **Using Two Underscore Characters in a Parameter Name**
Parameter names for Web methods cannot begin with "__" (two underscore characters). Names that begin with two underscore characters may create Web message parts that are unsupported (unusable) by XLANG/s.
- **The Any Element and the anyAttribute Attributes Are Not Supported in Web Methods**
You cannot use the **any** element or **anyAttribute** attribute in the schema for a Web method.
- **Using XLANG/s Keywords**
A Web service name or a Web method name cannot be a keyword in an XLANG/s. If you use an XLANG/s keyword in the Web service name or Web method name, you will get a compilation error when you add the Web service. For a list of reserved words for the XLANG/s language, see the [XLANG/s Reserved Words](http://go.microsoft.com/fwlink/?LinkId=155765) (http://go.microsoft.com/fwlink/?LinkId=155765).
- **Required XLANG/s Support for Parameter Types**
Using non-XLANG/s supported Web method parameter types will cause compilation errors. For example, BizTalk Server does not support a parameter that consists of a single dimensional array of schema types. In addition, BizTalk Server does not support

multidimensional arrays. For a list of words that XLANG/s language reserves in BizTalk Server, see [XLANG/s Reserved Words](http://go.microsoft.com/fwlink/?LinkId=155765) (http://go.microsoft.com/fwlink/?LinkId=155765).

- **Avoiding Compilation Errors Caused by Adding Web References Containing C# Keywords or Identifiers**

When you use the **Add Web Reference** to add Web references to BizTalk projects, BizTalk Server converts the schema types that are required to call each Web method to schemas. BizTalk Server adds these schemas to Reference.xsd. If your schemas contain element names that are C# keywords or the element name is not valid as a C# identifier, you may get a run-time error. To avoid run-time errors, ensure that the Web service you consume does not contain element names that are C# keywords or invalid C# identifiers.

- **Multiple Service/Port Type Definitions Are Unsupported**

BizTalk Server supports adding a Web service file with a single service and port type definition. If you add a WSDL file with multiple service or port type definitions, you may receive the following error:

```
Could not generate BizTalk files. Object reference not set to an instance of an object.
```

- **Support for Consuming Arrays Exposed by Web Services**

BizTalk Server can consume one dimensional and jagged arrays exposed by Web services that are not BizTalk Server Web services. For more information about how to consume Web service arrays, see [How to Consume Web Service Arrays](http://go.microsoft.com/fwlink/?LinkId=155766) (http://go.microsoft.com/fwlink/?LinkId=155766).



Note

Multi dimensional array syntax is not supported. For example, *MyArray[1,5]*.



Note

BizTalk Server does not support consuming an array of **DataSet** objects exposed by a Web service. The XLANG/s subservice does natively support the .NET DataSet class, but if you create a BizTalk project that contains a Web reference to a Web service that exposes an array of .NET DataSet objects you will get errors when you attempt to compile the project.

- **Web Method Parameters Must be Xml Serializable**

All parameters in a consumed Web service must be Xml Serializable. If you add a Web method that contains a parameter that is not Xml Serializable, you may receive the following error message:

```
System.Xml.Element must be Xml Seralizeable to be a message part type.
```



Note

The data types, **XmlDocument** and **DataSet**, while not Xml Serializable, are supported.

- **Consuming Messaging-Only Web Services**

When consuming messaging-only Web services, all BizTalk Server message body part names must match the Web method parameter names. For example, if the signature of the Web service is `WebMethod(MyType1 type1, MyType2 type2)`, the part name must be `type1` and `type2`, you may get the following runtime error:

```
Failed to retrieve the message part for parameter %1
```

For more information, see [How to Consume Web Services in a Messaging-Only Scenario](http://go.microsoft.com/fwlink/?LinkId=155767) (<http://go.microsoft.com/fwlink/?LinkId=155767>).

- **Configuring a SOAP Send Port Programmatically**

It is possible to set configuration properties programmatically on the message context. You can set these properties in an orchestration or a custom pipeline component whether the send port is static or dynamic.



Note

To configure the **MethodName** property for the static SOAP send port programmatically, you need to set **Method name** to **[Specify Later]** in the **Web Service** tab of the **SOAP Transport Properties** dialog box in the BizTalk Server Administration console.

For more information about the **MethodName** property, see [How to Dynamically Set the URI of a Consumed Web Service](http://go.microsoft.com/fwlink/?LinkId=155768) (<http://go.microsoft.com/fwlink/?LinkId=155768>).

- **Property Rules**

If the configuration property is set in an orchestration or in a custom pipeline component in a receive pipeline, then the following rules are applied:

- If a message is sent to a static send port, the property value will be overwritten with the value configured for that send port.
- If a message is sent to a dynamic send port, the property value will not be overwritten.

If a configuration property is set in a custom pipeline component in a send pipeline, then the following rule is applied:

- The value will not be overwritten regardless of whether the message is sent to a static or dynamic send port. In other words, send pipeline components overwrite the configuration property no matter where it was set.
- For more information about custom pipeline components, see [Developing Custom Pipeline Components](http://go.microsoft.com/fwlink/?LinkId=155769) (<http://go.microsoft.com/fwlink/?LinkId=155769>).
- For more information about the configuration properties of the SOAP send adapter, see [How to Dynamically Set the URI of a Consumed Web Service](http://go.microsoft.com/fwlink/?LinkId=155768) (<http://go.microsoft.com/fwlink/?LinkId=155768>).

- **Adding a Web Reference to a Consumed Web Service That Contains a Multi-Rooted Schema Will Cause a Compilation Error**

If you add a Web reference to your project for a Web service that was derived from a published BizTalk orchestration and the orchestration contains a schema with multiple roots, then an error will occur when the project is compiled. If you add a Web reference to your

project that was derived from a published BizTalk orchestration, ensure that the orchestration does not contain any multi-rooted schemas.

- **Using TypedDataSets as Parameters to Web Methods**

The following is what you need to do to support using TypedDataSets as parameters to Web methods:

- a. Add the Web reference to a C# project and then generate the proxy.
- b. Create a SOAP send port and specify the proxy on the send port and choose the method.
- c. In the orchestration, define a late bound port and define the message types. For most cases where no property promotion or distinguished field access is needed, the type can be defined as **XMLDocument**. Select PassThrough pipelines with this type.
- d. In the BizTalk Server Administration console, in the **Web Service** tab in the **SOAP Transport Properties** dialog box of the SOAP send port, specify that you want to use that proxy that you created. You will also need to specify assembly, type, and method.

- **Adding a Web Reference to a Consumed Web Service that Contains a Web Method Expecting Generic-Based Parameters Will Cause a Compilation Error**

If you add a Web reference to your project for a Web service that contains a Web method expecting generic-based parameters such as nullable parameters, an error will occur when the project is compiled. This is not supported. You must use explicit specialization to call the generic class from XLANG/s.

- **BizTalk Schema Generation Using the Add Web Reference**

When you use the **Add Web Reference** to add Web references to BizTalk projects, BizTalk Server converts the schema types that are required to call each Web method to schemas. BizTalk Server adds these schemas to Reference.xsd. To ensure that the **Add Web Reference** generates the BizTalk schemas correctly, the Web services must conform to the following guidelines:

- Web methods should have **SoapDocumentMethodAttribute** instead of **SoapRpcMethodAttribute**.
- Web services and methods must use the **Literal** binding instead of **Encoded** such as **[SoapDocumentMethod(Use=SoapBindingUse.Literal)]**.
- Web method parameters and return types must have **XmlRootAttribute** with a valid **Namespace** property unless they are native XSD types and the XmlNode type.
- Web methods must not use the **RequestNamespace** and **ResponseNamespace** properties in **SoapDocumentMethodAttribute**.
- Web services must be compliant with the Web services interoperability (WSI) Basic Profile version 1.1.

- **The Add Web Reference Does Not Support the Web Services Description Language (WSDL) Import Element**

The Add Web Reference fails when you add Web references for the WSDL file with the import element.

See Also

[Planning the BizTalk Server Tier](#)

Planning for Performance

BizTalk Server is an application platform. It is not just a server product or just a developer product. It is an application platform that is used to build business process management systems, to integrate enterprise applications, to automate workflows, and to enable service oriented architectures.

BizTalk Server is dependent on many other software components. The BizTalk application platform typically includes several of the following software components: the Windows Server operating system, SQL Server, BizTalk Server, IIS (optional), external systems that BizTalk Server is interacting with, as well as non-Microsoft adapters and components.

Because of the inherently complex nature of a BizTalk Server environment, there are many considerations to be made when planning for performance. There are several default settings that are applicable to all BizTalk Server environments and there are additional considerations and methodologies for optimizing specific BizTalk Server architectures.

This topic provides an overview of the default settings that you should apply when optimizing performance for all BizTalk Server environments. It also provides recommendations for testing and optimizing BizTalk Server environments that are designed for specific scenarios.

Settings That You Should Apply to All BizTalk Server Environments

The [Operational Readiness Checklists](#) section of this guide contains a list of items that you should review before employing any BizTalk solution. This checklist contains action items that can have a significant impact on the performance of your BizTalk Server environment regardless of the specific nature of the BizTalk solution that is employed.

Considerations for Testing and Optimizing a BizTalk Solution

Different BizTalk solutions may have drastically different performance criteria. For example, a BizTalk solution that is built around running orchestrations will have a different performance profile than a BizTalk solution that is focused on receiving, converting, and mapping flat file documents. An orchestration-focused solution may be CPU intensive or may call custom components that benefit from optimization, while a flat file conversion and mapping-focused solution may be more memory-intensive.

The adapters and pipelines used to receive and send documents in and out of BizTalk Server can also have a profound impact on the performance of the BizTalk solution. The level of document tracking required by the solution will also greatly impact performance. Because of the many divergent performance profiles that are possible in different BizTalk solutions, it is absolutely critical that that you test the BizTalk solution to measure maximum sustainable performance and maximum sustainable tracking performance.

After determining the maximum sustainable performance and maximum sustainable tracking performance of the BizTalk solution, there are specific steps that you can employ to remove

bottlenecks in the BizTalk solution. For more information, see the [BizTalk Server 2009 Performance Guide](http://go.microsoft.com/fwlink/?LinkID=150492) (http://go.microsoft.com/fwlink/?LinkID=150492).

See Also

[Planning the BizTalk Server Tier](#)

Planning for High Availability and Disaster Recovery

Solutions developed with BizTalk Server are often mission-critical enterprise-level applications that require maximum availability. When these solutions are placed into production, costs associated with downtime can be measured in thousands of dollars per second. For this reason, you should employ specific strategies to maximize the high availability and disaster recovery capabilities that are available with BizTalk Server and the dependency software and hardware required to support a BizTalk Server solution.

Hardware Considerations

To ensure high availability for a BizTalk Server environment, consider the following when planning for hardware:

- Plan on running multiple (at least two) BizTalk servers in a BizTalk group to accommodate running multiple instances of BizTalk Hosts across the BizTalk servers in the group. This will accommodate load balancing and fault tolerance of processes running in the host instances.
- Consider implementing a storage area network (SAN) to house the BizTalk Server databases. The SAN disks should be configured using RAID 1+0 (a stripe of mirror sets) topology if possible for maximum performance and high availability. For more information about using a SAN to house the BizTalk Server databases, see the [BizTalk Server Database Optimization white paper](http://go.microsoft.com/fwlink/?LinkID=101578) (http://go.microsoft.com/fwlink/?LinkID=101578).
- Plan on installing multiple SQL servers to house the BizTalk Server databases. Multiple SQL servers are required for SQL Server clustering (recommended) and/or housing certain BizTalk Server databases on separate physical SQL Server instances (also recommended).
- Consider using a virtual environment to control the hardware costs. Microsoft offers a range of virtualization products such as Microsoft Virtual Server 2005 R2, Windows Server 2008 Hyper-V, and Microsoft Hyper-V Server 2008. For recommendations on using BizTalk Server in a virtual environment, see [BizTalk Server 2009 Hyper-V Guide](http://go.microsoft.com/fwlink/?LinkId=151834) (http://go.microsoft.com/fwlink/?LinkId=151834).
- Plan on installing one or more Windows servers in a perimeter network domain to provide Internet-related services for your organization. Configure multiple Windows servers in the perimeter network domain using a network load balancing (NLB) solution. For more information, see [Network Load Balancing Deployment Guide](http://go.microsoft.com/fwlink/?LinkId=153139) (http://go.microsoft.com/fwlink/?LinkId=153139).

For more information about installing servers in a perimeter network, see [Configuring Your Domain When Exposing Transports to the Internet](#).



Note

A perimeter network is also known as a DMZ, demilitarized zone, and screened subnet.

Software Considerations

To ensure high availability for a BizTalk Server environment, consider the following when planning for software:

- Consider investing in the Enterprise Edition of BizTalk Server to accommodate scenarios that would benefit from clustering of BizTalk Hosts or that would benefit from running multiple MessageBox databases. Typically, the only reason that you should cluster a BizTalk Host would be to provide high availability for certain BizTalk adapters. For more information about providing high availability for BizTalk adapters using host clustering, see [Considerations for Running Adapter Handlers within a Clustered Host](http://go.microsoft.com/fwlink/?LinkID=151284) (<http://go.microsoft.com/fwlink/?LinkID=151284>).
- Plan on implementing a Windows Server cluster to house the BizTalk Server databases and the Enterprise Single Sign-On master secret server. For more information about using Windows Server cluster to provide high availability for the BizTalk Server databases and the Enterprise Single Sign-On master secret server, see [High Availability for Databases](#) and [High Availability for the Master Secret Server](#).

High Availability vs. Disaster Recovery

There are two prescribed methods for increasing availability for a BizTalk Server environment: providing high availability using fault tolerance and/or load balancing, or providing increased availability using disaster recovery. While each method increases availability, a key difference between them is that fault tolerance and/or load balancing typically provide much faster recovery time than disaster recovery. Fault tolerance and/or load balancing provide **high availability** whereas disaster recovery provides **increased availability**. For more information about implementing disaster recovery, see [Checklist: Increasing Availability with Disaster Recovery](#) and [Disaster Recovery](#).

See Also

[Increasing Availability for BizTalk Server](#)

[Checklist: Providing High Availability with Fault Tolerance or Load Balancing](#)

Planning for Testing

BizTalk Server testing can be divided into several categories including unit testing, functional testing, load testing, and availability testing. This topic describes unit and load testing and how to plan for each.

Planning for Unit Testing

Unit testing is a procedure used to validate that individual units of code are working as designed. Unit testing can be thought of as "break/fix" testing: Does the software perform the desired functionality under different conditions and can the software handle errors that occur under these conditions?

Since unit testing is typically performed on individual components, the associated test bed does not need the processing capabilities of an actual production environment. For this reason, you should consider performing unit testing in a Virtual Server environment, which requires considerably fewer hardware resources.

Another aspect of unit testing that may be performed in a virtualized environment is staging. Staging is the process of unit testing the actual deployment of a BizTalk solution. To maximize available hardware resources, consider using Virtual Server for your staging environment.

For more information about using BizTalk Server in a virtual environment, see [Using Virtual Server During the Release Management Process](#). For information about tools that may be useful for unit testing a BizTalk solution, see [Tools for Testing](#). For a checklist of considerations for performing unit testing see [Performing Unit Testing](#).

Planning for Load Testing

Load testing is the process of measuring maximum sustainable performance and maximum sustainable tracking performance of a BizTalk solution and then removing bottlenecks that inhibit the throughput of the solution. For more information about load testing and removing bottlenecks from a BizTalk Server solution, see the [BizTalk Server 2009 Performance Guide](#) (<http://go.microsoft.com/fwlink/?LinkID=150492>).

For information about tools that may be useful for load testing a BizTalk solution, see [Tools for Testing](#). For a checklist of considerations for load testing see [Performing Load and Throughput Testing](#).

Plan to Test for the Lifetime of the Solution

While unit testing and load testing are particularly important during the early stages of the solution, plan regular testing throughout the lifetime of the solution to uncover potential problems that may occur as load increases or as new functionality or components are added to the solution.

See Also

[Planning the BizTalk Server Tier](#)

[Checklist: Testing Operational Readiness](#)

BizTalk Server 64-Bit Support

For frequently asked questions related to 64-bit support for Microsoft BizTalk Server 2010, see [BizTalk Server 64-bit Support](#) (<http://go.microsoft.com/fwlink/?LinkID=155306>).

Planning the Database Tier

This section contains topics with planning recommendations for the database tier of a BizTalk solution.

In This Section

- [Planning for Database Performance](#)
- [Planning for Database Availability](#)
- [Planning for Database Testing](#)

Planning for Database Performance

Microsoft BizTalk Server is an extremely database intensive application that may require the creation of up to 13 separate databases in Microsoft SQL Server. Because of the database intensive nature of BizTalk Server, it is of critical importance that you choose the appropriate version and edition of SQL Server that will house the BizTalk Server databases. To optimize the performance of the computers running SQL Server that house the BizTalk Server databases, follow the recommendations in this topic and in the [BizTalk Server Database Optimization](#) white paper (<http://go.microsoft.com/fwlink/?LinkID=101578>).

BizTalk Server 2010 databases must be installed on either SQL Server 2008 SP1 or SQL Server 2008 R2.



Note

While this white paper is written for other versions of BizTalk Server and SQL Server, you can use the same recommendations for BizTalk Server 2010 and SQL Server 2008 SP1 / SQL Server 2008 R2.

Considerations for SQL Server Editions

BizTalk Server databases should be configured to run on a dedicated SQL Server instance whenever possible. BizTalk Server is a database intensive application, so separation of the BizTalk Server computers and the SQL Server computers that house the BizTalk Server databases will improve performance and should be considered a best practice in a production BizTalk Server environment.

Various editions of SQL Server provide different features which can affect the performance of your BizTalk Server environment. For example, under high-load conditions, the number of available database locks that are available for the 32-bit version of SQL Server may be exceeded, which is detrimental to the performance of the BizTalk solution. Consider housing your MessageBox database on a 64-bit version of SQL Server if you are experiencing "out of lock" errors in your test environment. The number of available locks is significantly higher on the 64-bit version of SQL Server.

Consider the table below when deciding on the database engine features that you will need for your BizTalk environment. For small-scale solutions, for example when running BizTalk Server 2010 Branch Edition, SQL Server 2008 SP1 / SQL Server 2008 R2 Workgroup Edition may be adequate for housing the BizTalk Server databases. For large scale, enterprise-level solutions that require clustering support, BizTalk log shipping support, or Analysis Services support, then you will need SQL Server 2008 SP1 / SQL Server 2008 R2 Enterprise Edition to house the SQL Server databases.

Version and Edition of SQL Server	64-bit support	Multi-Instance Support	Clustering support	Analysis Services
SQL Server 2008 SP1 / SQL Server 2008 R2 Enterprise Edition	Yes	Yes	Yes	Yes
SQL Server 2008 SP1 / SQL Server 2008 R2 Standard Edition	Yes	Yes	Yes (2 node)	Yes
SQL Server 2008 SP1 / SQL Server 2008 R2 Workgroup Edition	Yes	Yes	No	No



Note

BAM RTA requires SQL Server 2008 SP1 / SQL Server 2008 R2 Enterprise Edition.

For a complete list of the features supported by the editions of SQL Server 2008 R2, see

[Features Supported by the Editions of SQL Server 2008 R2](#)

(<http://go.microsoft.com/fwlink/?LinkId=151940>) in the SQL Server 2008 R2 documentation.

Planning for Database Availability

The BizTalk Server Messaging engine ensures that business processes are reliable and durable by persisting process state and business data to a SQL Server database known as the BizTalk Messagebox database. Because the reliability and durability of the persisted data is only as good as the underlying data store, planning for high availability of the BizTalk Server databases is critically important.

Hardware Considerations

To ensure high availability for the BizTalk Server databases, consider the following when planning for hardware:

1. Consider implementing a storage area network (SAN) to house the BizTalk Server databases. The SAN disks should be configured using RAID 1+0 (a stripe of mirror sets) topology if possible for maximum performance and high availability. For more information about using a SAN to house the BizTalk Server databases see the [BizTalk Server Database Optimization](#) white paper (<http://go.microsoft.com/fwlink/?LinkID=101578>).
2. Plan on installing multiple computers running SQL Server to house the BizTalk Server databases. Multiple computers running SQL Server will be required for SQL Server clustering (recommended) and/or housing certain BizTalk Server databases on separate physical SQL Server instances (also recommended).

3. Plan on installing one or more computers running SQL Server to implement SQL Server log shipping for purposes of disaster recovery. BizTalk Server implements database standby capabilities through the use of SQL Server log shipping. SQL Server log shipping automates the backup and restore of database and transaction log files, allowing a standby server to resume database processing in the event that the production server fails. For more information about implementing SQL Server log shipping for purposes of disaster recovery, see [What Is BizTalk Server Log Shipping?](#)

Software Considerations

To ensure high availability for the BizTalk Server databases, consider the following when planning for software: Plan to install a version and edition of SQL Server that supports failover clustering support and/or BizTalk log shipping support. For a complete list of the features supported by the editions of SQL Server 2008 SP1, see [Features Supported by the Editions of SQL Server 2008](#) (<http://go.microsoft.com/fwlink/?LinkId=151940>) in the SQL Server 2008 SP1 documentation.

High Availability vs. Disaster Recovery

There are two distinct methods for increasing availability for a BizTalk Server environment: providing high availability using fault tolerance and/or load balancing, or providing availability using disaster recovery. While each method increases availability, a key difference between them is that fault tolerance and/or load balancing typically provide much faster recovery time than disaster recovery does. Therefore, a solution built on fault tolerance or load balancing is more commonly thought of as providing high availability as opposed to merely providing availability. Both methods should be employed in a production BizTalk Server environment.

Provide high availability for the BizTalk Server databases using fault tolerance with Windows Clustering. For more information about providing high availability for the BizTalk Server databases, see [High Availability for Databases](#).

Increase availability with disaster recovery using BizTalk Server log shipping. To increase availability of the BizTalk Server databases with disaster recovery, follow the steps in the topic [Checklist: Increasing Availability with Disaster Recovery](#).

Planning for Database Testing

Thorough stress/load testing of a BizTalk solution figures prominently in the ultimate success or failure of the solution. Since BizTalk Server performance is so dependent on the performance of the BizTalk Server databases, testing and optimization of a BizTalk solution frequently focuses on testing and optimization of the computers running SQL Server that house the BizTalk Server databases.

Considerations When Planning for Database Testing

Consider the following when planning for database testing:

1. **Ensure that the test environment matches the production environment as closely as possible.** Using a virtual environment such as Microsoft Hyper-V Server 2008 for unit testing

is perfectly acceptable; however, all load/stress testing should be performed against hardware that matches the final production environment as closely as possible.

2. **Plan to measure maximum sustainable throughput and maximum sustainable tracking throughput of the BizTalk Server system.** Maximum sustainable throughput is measured as the highest load of message traffic that the BizTalk Server system can handle indefinitely in production. Follow the steps in the following topics in BizTalk Server 2010 Help:

- [Measuring Maximum Sustainable Engine Throughput](http://go.microsoft.com/fwlink/?LinkID=154388) (http://go.microsoft.com/fwlink/?LinkID=154388).
- [Measuring Maximum Sustainable Tracking Throughput](http://go.microsoft.com/fwlink/?LinkID=153815) (http://go.microsoft.com/fwlink/?LinkID=153815).

These topics describe the methodology for generating load against the system, the parameters that should be measured, and general recommendations to follow when testing MST.

3. **Understand the implications of how BizTalk Server implements host throttling.** The BizTalk Server host throttling algorithm attempts to moderate the workload of BizTalk Server host instances to ensure that the workload does not exceed the capacity of the host instance or any downstream host instances. The throttling mechanism is self tuning and the default configuration options are suitable for the majority of BizTalk Server processing scenarios. You should, however, have a good understanding of the throttling mechanism before implementing load/stress testing. For more information, see [Optimizing Resource Usage Through Host Throttling](http://go.microsoft.com/fwlink/?LinkId=155770) (http://go.microsoft.com/fwlink/?LinkId=155770) in BizTalk Server 2010 Help.

Planning the Development, Testing, Staging, and Production Environments

This topic discusses the environments used in the release management process for a BizTalk solution. As with any enterprise software solution, you should follow established software release management guidelines when you develop and release a BizTalk solution. This process should include the following distinct stages:

- Development
- Testing
- Staging
- Production

Ideally, you should complete each stage in the release management process in a discrete environment, separate from the other environments. Realistically, you may have to combine one or more of the environments due to hardware, time, or other resource constraints. At a bare minimum you should separate the production environment from the other environments.



Note

The latest installation and upgrade instructions for BizTalk Server 2010 are available as separate downloads on the [BizTalk Server 2010 Documentation page](http://go.microsoft.com/fwlink/?LinkID=194815) (http://go.microsoft.com/fwlink/?LinkID=194815).

Using Virtual Server During the Release Management Process

Consider completing development, unit testing, and staging in a "virtual" environment. Microsoft offers a range of virtualization products such as Microsoft Virtual Server 2005 R2, Windows Server 2008 Hyper-V, and Microsoft Hyper-V Server 2008. [Microsoft Virtual Server 2005 R2](http://go.microsoft.com/fwlink/?LinkId=71365) (<http://go.microsoft.com/fwlink/?LinkId=71365>) is available as a free download. Performing development work, unit testing, and staging in a virtual environment offers great flexibility and uses considerably fewer hardware resources than required otherwise. If a virtual environment is used, allocate at least 512 MB of memory for each virtual machine that is running on the host computer and an additional 512 MB of memory for the host operating system.

For example, for a BizTalk Server environment that uses five virtual machines (two computers running BizTalk Server, two Microsoft SQL Server cluster nodes, and one domain controller), you would plan to have 3 GB of memory installed on the host computer. If the BizTalk Server environment requires more than 2 GB of memory, consider installing a 64-bit version of Windows on the host computer to ensure that the maximum amount of installed memory is accessible by the host operating system.



Note

For recommendations on using BizTalk Server in a virtual environment, see [BizTalk Server 2009 Hyper-V Guide](http://go.microsoft.com/fwlink/?LinkId=151834) (<http://go.microsoft.com/fwlink/?LinkId=151834>).



Note

BizTalk Server 2010 is fully supported on a supported operating system that is running on any of the virtualization software listed in the Microsoft Knowledge Base Article 842301 [Microsoft BizTalk Server supportability on a virtual machine](http://go.microsoft.com/fwlink/?LinkId=158861) (<http://go.microsoft.com/fwlink/?LinkId=158861>). However, BizTalk Server 2010 may not perform as expected if installed on a supported operating system that is running in a virtualization software other than the ones mentioned in the KB article.

Development Environment

The BizTalk projects that are used for the BizTalk solution are created in the development environment. You should install the following software on the computers used in the BizTalk Server development environment:

- Internet Information Services (IIS)
- Visual Studio 2010
- SQL Server 2008 SP1 Client Tools
- BizTalk Server (including the following components)
 - Documentation
 - Administrative tools
 - Developer tools and SDK
 - Additional software
- SQL Server, if the BizTalk Server databases are to be hosted locally during development.

- Typically developers should have their own development computer (physical or virtual) with the necessary software installed.



Note

We recommend that you purchase and use MSDN subscription licenses for non-production environments. MSDN subscription licenses are offered at a significant discount from the cost of a retail license for the same software. For more information about obtaining an MSDN subscription see [MSDN Subscriptions](http://go.microsoft.com/fwlink/?LinkID=96006) (<http://go.microsoft.com/fwlink/?LinkID=96006>).

Testing Environment

Unit testing can be completed in a virtual environment. You should, however, conduct your performance testing in a physical environment with hardware and software that is identical to the production environment.

The testing environment is used to measure performance characteristics such as maximum sustainable throughput (MST) and maximum sustainable tracking throughput of the BizTalk solution. It should therefore match the physical production environment as closely as possible. For more information about measuring performance characteristics of a BizTalk solution see [Engine Performance Characteristics](http://go.microsoft.com/fwlink/?LinkID=155771) (<http://go.microsoft.com/fwlink/?LinkID=155771>) or the [BizTalk Server 2009 Performance Optimization Guide](http://go.microsoft.com/fwlink/?LinkID=150492) (<http://go.microsoft.com/fwlink/?LinkID=150492>).

Staging Environment

You typically use the staging environment to "unit test" the actual deployment of the BizTalk solution. The software installed in the staging environment should closely match the software installed in the production environment. It may, however, be acceptable to use virtual computers in the staging environment since this environment is not to be used for measuring performance. For more information about deploying a BizTalk application to a staging environment see [Staging Tasks for BizTalk Application Deployment](http://go.microsoft.com/fwlink/?LinkID=154686) (<http://go.microsoft.com/fwlink/?LinkID=154686>).

Production Environment

The production environment is the "live" environment that will host the running BizTalk solution. The production environment is the final endpoint in the release management process and should only host BizTalk applications that have previously undergone development, unit testing, load testing, and staging in the other environments. Thorough unit testing, load testing, and staging beforehand will help ensure maximum performance and uptime for the BizTalk application in the production environment.

Guidelines for Allocating Servers

The following guidelines provide a rule of thumb for the number of BizTalk servers and SQL servers you should allocate to each stage in the release management process given a particular

number of physical computers expected to be used in production: They are rough estimates which are subject to change depending on your architecture.



Note

Virtual servers may be used in the development and in the staging environment and can also be used for unit testing. All performance testing should be performed on physical hardware that matches the physical hardware in the production environment.

Computers running BizTalk Server used in production (physical hardware recommended)	Development servers (virtual or physical hardware)	Testing servers (physical hardware recommended)	Staging servers (virtual or physical hardware)	Total no. of computers running BizTalk Server
1	2	1	1	5
2	2	2	1	7
3	2	3	1	9
4	2	4	1	11

Estimated no. of computers running SQL Server used in production (physical hardware recommended)	Development servers (virtual or physical hardware)	Testing servers (physical hardware recommended)	Staging servers (virtual or physical hardware)	Total no. of computers running SQL Server
1	1	1	1	4
2	1	2	1	6
3	2	3	1	9
4	2	4	1	11

See Also

[Planning the Environment for BizTalk Server](#)

Managing BizTalk Server

This section provides guidance for managing a BizTalk Server 2010 system in a production environment. The procedures that follow generally assume you are working with a BizTalk

application after deployment. These procedures describe best practices and concepts for routine operations including the following:

- Checklists and procedures for deploying a BizTalk application into a production environment.
- Checklists and procedures for updating a BizTalk application after deployment to a production environment.
- Procedures to install, configure, and manage digital encryption certificates.
- Procedures for moving BizTalk Server databases from one instance of SQL Server to another.

In This Section

- [Managing Applications](#)
- [Managing Certificates](#)
- [Moving Databases](#)

Managing Applications

A BizTalk application is a logical container for application artifacts, such as orchestrations, pipelines, schemas, maps, and ports. BizTalk applications enable you to include related components in the same container. Any artifact within an application may refer to any other artifacts within that application, or to any artifact in any referenced application. For a complete list of artifacts that can be in a BizTalk application, see [What is a BizTalk Application?](http://go.microsoft.com/fwlink/?LinkId=154994) (<http://go.microsoft.com/fwlink/?LinkId=154994>).

BizTalk applications streamline many everyday BizTalk Server tasks. You can deploy, manage, start, stop, and troubleshoot BizTalk Server at the application level. This results in less confusion and less risk of error for users. A BizTalk application container contains

BizTalk Server assemblies that are deployed to it by the Visual Studio environment. The application may also contain receive ports, receive locations, send ports, property tracking settings, and role links. You can manually add BizTalk Server assemblies to the application, or move BizTalk Server assemblies from other applications. In addition, you can add non-BizTalk Server assemblies and BizTalk Server artifacts such as Business Rule Engine (BRE) policies and BAM definition files. Implicitly, the application also contains all of the bindings that are represented by their current settings.

You can create pre- or post-processing scripts to perform actions when an application is imported, installed, or uninstalled. For more information about using such scripts, see [Using Pre- and Post-processing Scripts to Customize Application Deployment](http://go.microsoft.com/fwlink/?LinkId=154995) (<http://go.microsoft.com/fwlink/?LinkId=154995>).

This section describes how to deploy, version, and update applications or individual artifacts.

In This Section

- [Deploying an Application](#)
- [Updating an Application](#)

Deploying an Application

Deployment is the logistical distribution of application artifacts to ensure all necessary components are available to the systems that require them. These artifacts include BizTalk Server assemblies, .NET assemblies, schemas, maps, bindings, business rules, and certificates. The BizTalk application can be leveraged to help roll out artifacts to other computers added to the group or for staging (when transporting an application to another environment).

There are many ways to deploy BizTalk artifacts for example importing them as part of an application by using the Deployment Wizard (from an .msi file), importing them using BTSTask.exe, deploying them from Visual Studio, or using MSBuild. If you import them using BTSTask.exe or deploy them from Visual Studio, you can either specify an application to deploy them to, or leave the application name blank, in which case they will be deployed to the default application.

Deploying by Using an .msi File

With BizTalk Server, you can deploy an application and its artifacts by exporting them to an .msi file. (For more information about applications and artifacts, see [What is a BizTalk Application?](http://go.microsoft.com/fwlink/?LinkId=154994) (<http://go.microsoft.com/fwlink/?LinkId=154994>). Deployment of a BizTalk application involves importing the application artifacts into the BizTalk management database as well as installing the artifacts on each computer that is member of the group. Deploying to an .msi file serializes all of the application artifacts into one package. This can be performed by doing an Export operation from the Administration console or by using BTSTask.exe from the command line. Once you have an .msi file, you can deploy all BizTalk Server assemblies to the BizTalk Management database for the group or run scripts specified to run at import time. This is done by using Microsoft Management Console (MMC) and doing an import operation of the .msi file (or via an import operation from the BTSTASK command line). The import operation of the .msi file creates the destination BizTalk application.

With the .msi file, you can deploy the application on a single computer, so that all BizTalk Server assemblies and dependency assemblies are stored in the Global Assembly Cache on the computer. The computer then has all of the binaries that it needs for run time. In this operation, you can also roll out Web services that are part of the solution, or apply computer-specific changes by scripts. This operation is performed by executing the .msi file. You can do this operation on each computer running BizTalk Server that is a member of the relevant BizTalk group. You can also use an .msi file to install an application on a new server added to a group.

If you are migrating your BizTalk application to a new group, you need to run the .msi installation on all servers in the new group. You need to import the .msi file once for the group. When you do, the application and its contents are installed on all run-time computers in the new group, and are also registered with the BizTalk Management database for the group. You can also add multiple binding files to an .msi file, in addition to the implicit bindings. Each additional binding file can be associated with an "environment". When multiple binding files are associated with a BizTalk Application during deployment, you can choose which binding files to use based on the environment (production, staging, or test) to which you are deploying.

You can use scripts with .msi files to customize a server (installation operation) or the group (import operation). For more information about using scripts with .msi files, see [Using Pre- and Post-processing Scripts to Customize Application Deployment](#) (<http://go.microsoft.com/fwlink/?LinkId=154995>).

For a checklist of steps for deploying an application, see [Checklist: Deploying an Application](#).

Exporting an Application's Bindings by Using a Binding File

With BizTalk Server, you can export the bindings of an application to a binding file, and then import those bindings from the binding file to another application. To do so, the destination application must already exist; the import procedure does not create the application. The binding file is an XML file that contains the bindings of all artifacts in an application, group, or assembly. You can also export all bindings for a BizTalk group, or the bindings for a BizTalk assembly. For more information about working with bindings, see [How to Export Bindings to a Binding File](#) and [How to Import Bindings from a Binding File](#).

You can also add a binding file as a resource to an .msi file. For more information about adding a binding file as a resource, see [How to Export an Application to an .msi File](#).

For more information about application deployment in general, see [Understanding BizTalk Application Deployment and Management](#) (<http://go.microsoft.com/fwlink/?LinkId=154996>).

In This Section

- [Best Practices for Deploying an Application](#)
- [Known Issues with Deploying an Application](#)
- [Permissions for Managing an Application](#)
- [Artifacts That Must Be Unique in an Application](#)
- [Deploying and Exporting an Application](#)

Best Practices for Deploying an Application

This topic lists best practices that you should follow in deploying BizTalk applications.

Deploying a BizTalk Application

Document application deployment procedures

- Make sure that all the procedures used in application deployment are documented in depth, so you have a record of how the deployment was performed and will know how to deploy further or undeploy. Anything that is not scripted should be documented with detailed steps. This should include documenting any changes to external systems and deployment of third-party components.

Script application deployment

- Script as many application deployment steps as possible. Scripting reduces the risk of human error during the deployment process.

Creating a BizTalk Application

Script the creation of BizTalk application and .msi files

- BtsTask.exe can be used to script the creation of BizTalk applications. If the creation of the applications is scripted, then the packages can be automatically built using an automated process on a build server. For more information about scripting the creation of applications, see [Deploying and Managing BizTalk Applications](http://go.microsoft.com/fwlink/?LinkID=154210) (http://go.microsoft.com/fwlink/?LinkID=154210) and the [BizTalk Server 2006: Understanding BizTalk Server Application Deployment](http://go.microsoft.com/fwlink/?LinkID=101599) (http://go.microsoft.com/fwlink/?LinkID=101599) white paper.

**Note**

The white paper also applies to BizTalk Server 2010.

Deploying a BizTalk Assembly**Never deploy an assembly from Visual Studio on a production computer**

- During the development process, a developer must often redeploy assemblies from Visual Studio. To enable the redeployment, Visual Studio may undeploy, unbind, stop, and unenlist artifacts included in the assemblies. Although this is necessary and appropriate in the development environment, it can cause unexpected and undesired consequences in a production environment. For this reason, as well as to prevent anyone from deploying an assembly from Visual Studio on a production computer, we recommend that you never install Visual Studio on a production computer.
- In addition, never refer to a production database from a computer running Visual Studio.

Adding Artifacts to a BizTalk Application**Group related artifacts together in a single application**

- As much as possible, place related artifacts in the same BizTalk application. This allows you to manage and deploy the artifacts as a single entity, making management easier. You can group artifacts that support the same business process or artifacts that perform similar functions into a single application.

Deploy shared artifacts in a separate application

- If artifacts are going to be shared by two or more applications, deploy the shared artifacts into a separate application. For example, if two applications share a schema, place the schema in a separate application. We recommend this because only one artifact in a BizTalk group can have a single locally unique identifier (LUID). A LUID consists of the artifact name and optionally other attributes. If you include an artifact in one application, and then create a reference to it from another application, the referring application may not function correctly when you stop the application containing the artifact.

This best practice applies to all artifact types except for files, such as Readme files and scripts, which are added to the application as a File type of artifact. This is because more than one file artifact with the same name can be deployed in a BizTalk group. Therefore, you can use a file having the same name in two or more applications. In this case, stopping one application will not impact the other application. For more information about adding file artifacts, see [How to Add a File to an Application](http://go.microsoft.com/fwlink/?LinkId=154997) (http://go.microsoft.com/fwlink/?LinkId=154997).

Deploy a shared Web site in a separate application

- If a Web site will be shared by more than one business solution, deploy the Web site in a separate application. This is because when you uninstall a BizTalk application, the virtual directory of any Web site that is part of the application is removed, even if the Web site is running. If the Web site is shared with another business solution, the other business solution will no longer function correctly as a result.

Deploy shared policies in a separate application

- If a policy is used by two or more applications, you should deploy it in a separate application rather than creating a reference from one application to another. This is because when you stop an application, its policies are undeployed. If you stop an application that includes a policy used by another application, the policy will no longer function in either application.

Deploy shared certificates in a separate application

- If a certificate is used by a send port or receive location in two or more applications, you should deploy the certificate in a separate application, and then reference this application from the applications that need to use the certificate. This is because only one artifact in a BizTalk group can have a single LUID, so you will not be able to import the same certificate in two different applications. If you attempt to import two applications that each use the same certificate, the first import will succeed, and the second will not. In this case, using the Overwrite import option does not correct the problem, as the existing certificate that you want to overwrite is contained in another application.

Exporting and Importing a BizTalk Application

When deploying large .msi files, you may need to increase the default transaction timeout of the COM+ components used by BizTalk Server to deploy applications

- If you attempt to deploy an .msi file that is very large (more than 100 MB), then the application may not be deployed within the default transaction time-out of the COM+ components that are used by BizTalk Server during application deployment. If the transactions associated with these COM+ components time out before the deployment is completed, then the deployment will fail. If you are deploying very large .msi files, then consider taking one of these approaches to mitigate this problem:
- Deploy several smaller .msi files instead of one large .msi file.
 - Increase the default transaction timeout of 3,000 seconds associated with the Microsoft.BizTalk.ApplicationDeployment.Group and the Microsoft.BizTalk.Deployment.DeployerComponent components in the Component Services management interface. These components belong to the Microsoft.BizTalk.ApplicationDeployment.Engine and Microsoft.Biztalk.Deployment COM+ applications, respectively. For more information, see Microsoft Knowledge Base article 287499, [How to Change the Transaction Time-Out Value for MTS or COM+](http://go.microsoft.com/fwlink/?LinkId=109589) (<http://go.microsoft.com/fwlink/?LinkId=109589>).

Prevent bindings from being overwritten

- If you do not want the bindings in the application you are exporting to overwrite the bindings in an application into which you are importing the .msi file, then you should not select the binding file as a resource to export during the export operation.

Ensure that the .msi file is secure

- An .msi file may contain sensitive data. Be sure to take steps to help ensure that the file is secure. For more information about .msi file security, see [Security and Windows Installer](http://go.microsoft.com/fwlink/?LinkId=154998) (<http://go.microsoft.com/fwlink/?LinkId=154998>).

Ensure that the binding file is secure

- A binding file may contain sensitive data. Be sure to take steps to help ensure that the file is secure.

Schedule exports when no one is making changes to an artifact

- Schedule export operations during hours when users are not likely to be making changes to artifacts. This can be important because if a user is modifying a database-based artifact, virtual directory, certificate, or policy while an export operation is in progress, the changes will not be reflected in the exported .msi file.

Importing a BizTalk Application

Script the importing of .msi files

- BtsTask.exe can be used to script the importing of existing BizTalk .msi files. For more information about scripting .msi file importing, see [Deploying and Managing BizTalk Applications](http://go.microsoft.com/fwlink/?LinkId=154210) (<http://go.microsoft.com/fwlink/?LinkId=154210>) and the [BizTalk Server 2006: Understanding BizTalk Server Application Deployment](http://go.microsoft.com/fwlink/?LinkId=101599) (<http://go.microsoft.com/fwlink/?LinkId=101599>) white paper.

Note

The white paper also applies to BizTalk Server 2010.

- You can add scripts to run as pre-processing or post-processing scripts. However, you will have to include logic in your scripts to check environment variables to determine which context the script is executing in (an import, installation, or uninstallation) and process accordingly. For more information about using pre- and post-processing scripts, see [Using Pre- and Post-processing Scripts to Customize Application Deployment](http://go.microsoft.com/fwlink/?LinkId=154995) (<http://go.microsoft.com/fwlink/?LinkId=154995>).

Verify the existence of referenced artifacts

- When an application that you are importing has a reference to another application, BizTalk Server verifies that the referenced application exists. However, it does not verify that the artifacts on which your application has dependencies are included in the referenced application. When you import an application that has dependencies to artifacts in another application, we recommend that you verify that the referenced application contains the required artifact or artifacts.

Importing from an .msi file precludes storing changed assemblies in the global assembly cache

- To update the artifacts in an application, import the changed or updated artifacts from an .msi file into the application. If you do not use an .msi file to import the artifacts, you will need to update all servers in the group by storing the changed assemblies in the global assembly cache.

Host processing groups to update a subset of total servers

- When updating artifacts in an application, you must normally update all servers in a BizTalk group. However, if you use host processing groups, you only need to update a subset of the total servers in the group.

If an import operation times out, split the application into additional .msi files

- An import operation will time out if it exceeds 3,600 seconds in duration. If you are attempting to import an .msi file and the operation times out, you should divide the contents of the application into more than one .msi file by re-exporting the application and selecting a subset of artifacts to export. For more information about exporting an application into an .msi file, see [How to Export a BizTalk Application](http://go.microsoft.com/fwlink/?LinkID=154848) (http://go.microsoft.com/fwlink/?LinkID=154848).

Known Issues with Deploying an Application

Deploying a BizTalk Application

Deploying an updated application requires correcting references

If you deploy an entirely new application to replace an existing application, you must correct any references between other applications and the application that you are replacing.

Using Visual Studio to deploy an application can stop applications



Note

We recommend that you avoid using Visual Studio to deploy an application into a production environment.

If you use Visual Studio to deploy an application into a production environment and the Restart Host Instances option is set to True in project properties, all host instances will restart when you deploy the application, including those that are not associated with this application. This will stop all other applications that are running on any host instance on the local computer.

Adding Artifacts to a BizTalk Application

Moving an artifact can move dependencies

When you move an artifact to a new application, any other artifacts on which it has dependencies are also moved unless the new application has a reference to the application(s) containing the artifacts that the moved artifact depends upon. Also, any artifacts that have dependencies on the moved artifact also will be moved unless the application(s) containing them have a reference to the new application. When moving an artifact, you will be shown the list of other artifacts that will also be moved. For instructions on moving an artifact, see [How to Move an Artifact to a Different Application](http://go.microsoft.com/fwlink/?LinkId=154999) (http://go.microsoft.com/fwlink/?LinkId=154999).

Exporting a BizTalk Application

An incorrect error can be displayed when installing an .msi file on Windows Vista

When installing an .msi package exported using BizTalk Server 2006 on BizTalk Server 2010 running on Windows Vista®, you may receive the following incorrect error, "The installer has encountered an unexpected error installing this package. This may indicate a problem with this package. The error code is 2869." To correct this error, first import the package using BizTalk Server 2010 and then re-export and install the package.

Importing a BizTalk Application

Passwords are not removed from binding files added to an application

For security reasons, during application export, passwords are removed from application bindings. They are not, however, removed from any binding files that were added to the application. After importing the application, you will need to reconfigure the passwords in order for the application to function. You can do this by editing the binding file or by using the Administration console. For more information about editing a binding file, see [Customizing Binding Files](http://go.microsoft.com/fwlink/?LinkId=155000) (http://go.microsoft.com/fwlink/?LinkId=155000). For more information about configuring security for adapters, see [Using Adapters](http://go.microsoft.com/fwlink/?LinkId=155001) (http://go.microsoft.com/fwlink/?LinkId=155001).

BizTalk Server does not roll back scripted imports or installs

If an import fails, BizTalk Server rolls back all import operations except for any actions performed by custom scripts. This is also true for install and uninstall operations.

A missing schema might not be reported after an import

If you create a filter for a send port in one application that uses a property schema in another application, and then import the first application into a new BizTalk group, you will not receive a warning that the schema is missing, and filtering will not function when the application is installed and started. You can correct the problem by importing the application that contains the schema before you install the application that does not contain the schema.

See Also

[Deploying an Application](#)

Permissions for Managing an Application

For information about the permissions required for deploying and managing applications, see [Permissions Required for Deploying and Managing a BizTalk Application](http://go.microsoft.com/fwlink/?LinkId=154733) (http://go.microsoft.com/fwlink/?LinkId=154733).

Artifacts That Must Be Unique in an Application

For information about the unique artifacts in an application, see [Artifacts That Must Be Unique in an Application or Group](http://go.microsoft.com/fwlink/?LinkId=155141) (http://go.microsoft.com/fwlink/?LinkId=155141).

Deploying and Exporting an Application

This section contains detailed information about how to perform the steps involved in deploying a BizTalk application. The topics in this section support the following checklists:

- [Checklist: Deploying an Application](#), which shows how to deploy an application in a development environment, export it into an .msi file, and then import it into a production environment from the .msi file.
- [Checklist: Exporting Bindings from One Application to Another](#), which shows how to export bindings from an application into another application in either a development or production environment.

You can use the following tools to deploy and manage a BizTalk application:

Tool	Use
BizTalk Server Administration console	<p>From this graphical user interface, you can perform all of the deployment and management operations for a BizTalk application, including the following:</p> <ul style="list-style-type: none"> • Starting the Import, Installation, and Export Wizards • Adding and removing an application's artifacts, and making other modifications to the application • Generating BizTalk Server .msi files for a BizTalk application • Installing the application onto a computer from the .msi file or importing the application from the .msi file into another BizTalk group • Importing the bindings for an application from a binding file <p>For more information about the administration console, see Using the BizTalk Server Administration Console (http://go.microsoft.com/fwlink/?LinkID=154689).</p>
BTSTask command-line tool	<p>You can perform many application management tasks from the command line. For more information about the command-line tool, see BTSTask Command Line Reference (http://go.microsoft.com/fwlink/?LinkID=155003).</p>
Scripting and programmability APIs	<p>You can use Microsoft Windows Management Instrumentation (WMI) or the BizTalk Explorer Object Model to create and run scripts that automate many application management tasks. For more information about scripting, see WMI Class Reference (http://go.microsoft.com/fwlink/?LinkID=155004).</p>
Visual Studio 2010	<p>You can create BizTalk assemblies in Visual Studio, use the Deploy command to automatically deploy them into a BizTalk application, and use BizTalk Explorer to configure application artifacts from within Visual Studio. For more information about working within Visual Studio, see Deploying BizTalk Assemblies from Visual Studio into a BizTalk</p>

Tool	Use
	Application (http://go.microsoft.com/fwlink/?LinkID=154719).

In This Section

- [Creating an Application](#)
- [Deploying an Assembly](#)
- [Adding Artifacts to an Application](#)
- [How to Export an Application to an .msi File](#)
- [How to Export Bindings to a Binding File](#)
- [How to Add a Binding File to an Application](#)
- [How to Import an Application from an .msi File](#)
- [How to Install an Application](#)
- [How to Import Bindings from a Binding File](#)
- [Testing an Application](#)
- [How to Add a Reference to an Application](#)

Creating an Application

There are three ways to create a BizTalk application. There are also several options for naming, referencing, and deploying artifacts to applications.

Creating a BizTalk Application

The three ways are as follows:

1. Create the BizTalk application by using the New Application command of the BizTalk Server Administration console or the AddApp command of the BTSTask command-line tool. For more information about using these commands, see [How to Create an Application](#) (http://go.microsoft.com/fwlink/?LinkId=155007).
2. Import an .msi file that was exported from another application. If the application does not already exist in the current BizTalk group, the application, including its artifacts, is automatically created in the current BizTalk group. For more information about importing applications, see [How to Import a BizTalk Application](#) (http://go.microsoft.com/fwlink/?LinkId=154827).
3. Deploy BizTalk assemblies from Visual Studio into a BizTalk application. If the application specified in the deployment properties for a project in Visual Studio does not exist in the current BizTalk group, it will be automatically created. For more information about deploying an assembly from Visual Studio, see [Deploying BizTalk Assemblies from Visual Studio into a BizTalk Application](#) (http://go.microsoft.com/fwlink/?LinkId=154719).

To deploy an application using an .msi file, the destination application does not need to exist, but will be automatically created. However, to import bindings from one application to another, the destination application must already exist. If not, you need to create it by performing one of the procedures listed above.

For more information about the specific steps required to create an application, see [How to Create an Application](http://go.microsoft.com/fwlink/?LinkId=155007) (http://go.microsoft.com/fwlink/?LinkId=155007).

Application Options

Before creating a new application, you should do the following:

- Determine a name that is unique within the BizTalk group for the application.
- Add a reference from the new application to any application containing artifacts that you want to reuse in the new application. For more information about dependencies between applications, see [Dependencies and Application Deployment](http://go.microsoft.com/fwlink/?LinkId=155008) (http://go.microsoft.com/fwlink/?LinkId=155008).
- Determine whether you want to deploy some artifacts into separate applications, for example, shared artifacts that should be deployed into their own separate application.

Deploying an Assembly

Deploying an assembly builds the assembly and imports it, along with the orchestrations, pipelines, schemas, and maps (artifacts) that it contains into the local BizTalk Management database. Initially, this is done in the development environment.

Deployment also associates the assembly with the BizTalk application that you have specified in project properties within Visual Studio. After you deploy a solution, you can view and manage the deployed assemblies and their artifacts from within the BizTalk Server Administration console or by using the BTSTask command-line tool. You can manage the artifacts either individually or grouped within the application.

Deploying an Assembly

You can add assemblies to applications in the following ways:

- Deploy an assembly to an application from within the Visual Studio environment
- Manually add BizTalk Server assemblies to the application from within the BizTalk Server Administration console
- Add a BizTalk assembly to an application by using script from the command line
- Move BizTalk Server assemblies from other applications from within the BizTalk Server Administration console

For more information about adding assemblies to applications, see [Deploying BizTalk Assemblies from Visual Studio into a BizTalk Application](http://go.microsoft.com/fwlink/?LinkID=154719) (http://go.microsoft.com/fwlink/?LinkID=154719).

Redeploying Assemblies

In the process of developing and debugging your BizTalk assemblies, you may need to redeploy them multiple times. BizTalk Server provides a simple mechanism for redeployment. If you are redeploying an assembly without changing the version number, you can use the Redeploy property. BizTalk Server will automatically perform all of the steps to redeploy the assembly for you.

For more information about redeploying assemblies, see [How to Redeploy a BizTalk Assembly from Visual Studio](http://go.microsoft.com/fwlink/?LinkID=154720) (http://go.microsoft.com/fwlink/?LinkID=154720).

Best Practices for Redeploying an Assembly

You must install the new assembly in the GAC

- When you redeploy an assembly, you must always install the new version of the assembly in the global assembly cache (GAC). You can do this after you redeploy it. For more information, see [How to Install an Assembly in the GAC](http://go.microsoft.com/fwlink/?LinkID=154828) (<http://go.microsoft.com/fwlink/?LinkID=154828>).

You should always redeploy at the solution level when there are dependencies

- If you have multiple assemblies in a solution, and one or more assemblies in the solution has a dependency on the assembly that you want to redeploy, you should redeploy your assemblies at the solution level. This is because when you redeploy an assembly at the project level, BizTalk Server will stop, unenlist, unbind, and remove the artifacts in all assemblies that either depend on this assembly or on upon which this assembly depends. BizTalk Server will not take the additional steps to deploy, bind, enlist, and start the artifacts. When you redeploy the entire solution, however, BizTalk Server automatically takes the steps required to undeploy and redeploy all of the artifacts in the solution based on their dependencies.

You may need to manually redeploy dependent assemblies

- BizTalk Server always undeploys dependent assemblies when it undeploys an assembly, but in the following cases, you must take the additional steps to deploy, bind, and enlist the artifacts in each dependent assembly after redeploying the assembly on which the assembly depends:

If you redeploy an assembly at the project level and another assembly in the same solution depends on it.

If you redeploy an assembly at the solution level, but a dependent assembly exists in a different solution.

You must restart host instances

- When you redeploy an assembly that contains an orchestration without changing the assembly version number, the existing assembly is overwritten in the BizTalk Management database. Before the change will take effect, however, you must restart each host instance of the host to which the orchestration is bound. You can specify the option that all host instances on the local computer restart automatically when you redeploy an assembly.

When you redeploy an assembly that contains an orchestration without changing the assembly version number, the existing assembly is overwritten in the BizTalk Management database. Before the change will take effect, however, you must restart each host instance of the host to which the orchestration is bound. You can specify the option that all host instances on the local computer restart automatically when you redeploy an assembly. For more information about deployment properties, see [How to Set Deployment Properties in Visual Studio](http://go.microsoft.com/fwlink/?LinkID=154718) (<http://go.microsoft.com/fwlink/?LinkID=154718>).

You can also manually stop and start each host instance. For more information about stopping and starting a host instance, see [How to Stop a Host Instance](http://go.microsoft.com/fwlink/?LinkID=154829) (<http://go.microsoft.com/fwlink/?LinkID=154829>) and [How to Start a Host Instance](http://go.microsoft.com/fwlink/?LinkID=154830) (<http://go.microsoft.com/fwlink/?LinkID=154830>).

Adding Artifacts to an Application

You can add and configure artifacts, such as send and receive ports, receive locations, and orchestrations, by using the Administration console. You can also generate binding files and add them to the application if you want to apply different bindings for the different environments that you will import the application into.

You can add additional (typically non-BizTalk Server) artifacts, such as scripts, certificates, and Readme files, to the application under the Resources node. To do so, use the Administration console or BTSTask.

For more information about artifacts, see [How to Create or Add an Artifact](http://go.microsoft.com/fwlink/?LinkID=154724) (<http://go.microsoft.com/fwlink/?LinkID=154724>), [Managing Artifacts](http://go.microsoft.com/fwlink/?LinkID=154725) (<http://go.microsoft.com/fwlink/?LinkID=154725>) and [Binding Files and Application Deployment](http://go.microsoft.com/fwlink/?LinkID=154726) (<http://go.microsoft.com/fwlink/?LinkID=154726>).

Factoring Artifacts into Multiple BizTalk Applications

During the development process, you may have deployed your assemblies into a single application for convenience. For various reasons, you may want to factor the artifacts into multiple applications before they are deployed into production.

Before deploying, you should perform an in-depth analysis of the factoring of an assembly. Determine whether you should perform No factoring, Full factoring, or Optimal factoring.

No Factoring

All BizTalk artifacts are in one assembly. This is the easiest to understand and deploy, but provides the least amount of flexibility.

Full Factoring

Each BizTalk artifact is in its own assembly. This provides the most flexibility, but is the most complex to deploy and understand.

Optimal Factoring

Optimal factoring falls between No Factoring and Full Factoring based on in-depth analysis of your BizTalk applications. In addition to versioning, this allows you to more easily implement your BizTalk host design. This is achieved by looking for relationships among BizTalk artifacts. If possible, put artifacts that are always versioned together in the same assembly. If independent versioning of the artifacts is required, then they must be put in different assemblies. This is the level of factoring that you want to achieve.

How to Export an Application to an .msi File

You can use the Export MSI File Wizard or BTSTask to export the application artifacts into an .msi file that you will use to import the application into a new BizTalk group. This process will also install the application on the computers that will run it.

Exporting Application Bindings in an .msi File

When you export an application into an .msi file, you select which resources to export. These resources can include all or a subset of the deployed assemblies or bindings that are available for export.

To make it easier to import the applications back into your development environment at a later time to make changes or additions, you may want to export the bindings for each application and

then add them back into the applications. When you set the Resource type to "BiztalkBinding," you can add additional binding files to an .msi file. For each binding file that you add, you must specify the environment name (text field) that the bindings should be applied to. Then on import, you will be asked to select the environment name to which you are presently importing. If these match, then the binding file is applied to the target group. You can also specify no environment name for the added binding file, which will cause the set of bindings to be applied to all environments unconditionally.

Using an .msi file automates what can otherwise be a substantial set of manual tasks. In a production environment, you can make it easier to deploy an assembly into multiple BizTalk groups by transporting the bindings for the assembly along with the assembly. If your application contains a significant number of artifacts, you can export some of the artifacts into one Windows Installer package and some into one or more other Windows Installer packages.

While exporting the application, you should consider some important points. For example, you may either export all of the artifacts in the application or only the artifacts that you want to add or update. If you export a file artifact, make sure that it is the version that you want to use in the application. For more such points and for more information about exporting a BizTalk application into an .msi file, see [How to Export a BizTalk Application](http://go.microsoft.com/fwlink/?LinkID=154848) (<http://go.microsoft.com/fwlink/?LinkID=154848>).

How to Export Bindings to a Binding File

You can export the bindings of a BizTalk application into another existing BizTalk application using a binding file. You can also export all the bindings in a group or the binding for an assembly. Subsequently, you can import those bindings into either an application or group.

A binding file is an XML file that describes the artifacts in the BizTalk Management database and their relationships. It contains binding information for each BizTalk Server orchestration, pipeline, map, or schema in the scope of a BizTalk assembly, application, or group. It contains the configuration settings for each send port, send port group, receive port, receive location, and party. It also describes which host each orchestration is bound to and its trust level.

Why to Export to a Binding File

Binding files can speed deployment in the following scenarios by avoiding the need to manually configure bindings:

- **Moving an application from one deployment environment to another**
Using a binding file can speed deployment by avoiding the need to manually configure bindings for different deployment environments, such as from a development environment to a test environment.
- **Updating an assembly**
You can use a binding file to automatically apply or reapply the bindings to an assembly after an update of the assembly.
- **Deploying an assembly to multiple BizTalk groups**
You can avoid the need to separately configure the bindings for an assembly deployed into multiple BizTalk groups by using a binding file.

Using a binding file gives you flexibility in applying bindings to an application. When you export an application to an .msi file, you can only specify that all of the bindings for the application will be exported to the .msi file. With binding files you can do the following:

- Export to a binding file all bindings from the current application, all bindings from the current group, or only the bindings for a single assembly. (You do so by using the Export Bindings command for an application in the Administration console.)
- You can add a binding file to an application (using the Add Resources command) so that its bindings are applied immediately or so the bindings are applied when the application is imported into another group.
- You can add multiple binding files to an application (using the Add Resources command) and specify a target deployment environment for each one. This enables you to use a single deployment package for multiple deployment environments. When you import the application, you can select which bindings to apply.
- You can export separate binding files for multiple assemblies in an application.
- You can edit binding files after you generate them to change their binding information.

How to Export to a Binding File

You export the bindings of an application to a binding file by executing the Export Bindings command for the application in the BizTalk Server 2010 Administration console, or by using the BTSTask ExportBindings command on the command line.

For security reasons, when you export a binding file, BizTalk Server removes the passwords for the bindings from the file. After importing the bindings, you must reconfigure passwords for send ports and receive locations before they will function. You configure passwords in the Transport Properties dialog box of the BizTalk Server Administration console for the send port or receive location.

Information in a binding file supersedes existing configuration information. If the name of an artifact in a binding file matches the name of an artifact in your existing configuration, the artifact in the binding file will update the artifact in your existing configuration when you import the binding file.

For information about how hosts and trust levels are stored in binding files, how host and trust levels in a binding file are matched to hosts and trust levels in the application, and the order in which bindings are applied, see [Binding Files and Application Deployment](http://go.microsoft.com/fwlink/?LinkID=154726) (http://go.microsoft.com/fwlink/?LinkID=154726). For instructions on how to export bindings for a BizTalk application, see [How to Export Bindings for a BizTalk Application](http://go.microsoft.com/fwlink/?LinkId=155009) (http://go.microsoft.com/fwlink/?LinkId=155009).

How to Add a Binding File to an Application

For information about adding a binding file to an application, see [How to Add a Binding File to an Application](http://go.microsoft.com/fwlink/?LinkId=155010) (http://go.microsoft.com/fwlink/?LinkId=155010).

How to Import an Application from an .msi File

You can use the Import MSI Wizard in the BizTalk Server 2010 Administration Console or BTSTask to import a BizTalk application from an .msi file into a BizTalk group in the target

environment and install the application on individual host instances in the group. The full import process performs the following operations:

- A group-level deployment of the application
- A server-level installation of the application.

Group-Level Application Deployment

You perform a group-level deployment of an application on a server in the group by running the Import MIS Wizard from the BizTalk Server Administration console or by running BTSTask. The group-level deployment does the following:

- Creates the application and its artifacts in the group
- Imports bindings resident in the .msi package
- Deploys all BizTalk Server assemblies with their artifacts to the BizTalk Management database for the group
- Runs scripts specified to run at import time.

If you have added environment-specific binding files to the application, you will have to select the bindings that you want to apply on import.

Server-Level Application Installation

You perform a server-level installation of an application on each server in a group by double-clicking the .msi file itself, or performing the install process at the end of the Import MSI Wizard. Instead of being done once per group, it is typically done on each BizTalk server that is a member of the group. The server-level installation does the following:

- Installs all BizTalk Server assemblies and dependency assemblies into the global assembly cache of the server, so that this computer has all of the binaries that it needs for runtime
- Rolls out related Web services that might be part of the solution, for example, orchestrations published as Web services.
- Applies computer-specific changes, such as pre-creating MSMQ queues or creating FILE drop folder structures and permissions, which can be done with the help of scripts.

When you execute an .msi file to install an application, the .msi file creates registration entries in the Add or Remove Programs list, and accelerates deployment by automating deployment of artifacts and their dependencies in the correct order.

For more information on installing a BizTalk application, see [How to Install an Application](#).

The Complete Application Deployment and Installation Process

The Import MSI Wizard deploys the application on the group. It does not install the application on the individual servers in the group. If the application includes file-based artifacts, you need to install the application on each host instance that will run the assemblies in the application (and any computers running applications that depend on this application). You can do so on the server you ran the Import MSI Wizard on, however, by selecting the **Run the Application Installation Wizard to install the application on the local computer** check box on the Import Succeeded page displayed by the Import MSI Wizard. You can do so on the other servers in the group by double-clicking the .msi file on each of those servers.

If you are ready to test the application, you can import it into a BizTalk group in a test environment. If your application is ready for staging or production, you can import it into one of those environments.

Important Considerations

When importing a BizTalk application from an .msi file, keep the following in mind:

- **You must specify that you want artifacts to be overwritten in a standard import process.** If you want to overwrite existing artifacts, select the option to overwrite existing artifacts when importing the .msi file.
- **Imported bindings overwrite existing bindings.** When you import an .msi file that contains bindings into an existing application, the existing bindings are overwritten by imported bindings that have the same name. This is the case even when you have not selected the option to overwrite existing artifacts when importing the .msi file. If you do not want the bindings in the application you are exporting to overwrite the bindings in an application into which you are importing the .msi file, then you should not select the binding file as a resource to export during the export operation. For more information about setting the resources for an export, see [How to Export a BizTalk Application](http://go.microsoft.com/fwlink/?LinkID=154848) (<http://go.microsoft.com/fwlink/?LinkID=154848>).

As bindings are applied during the import process, bindings that have already been applied are overwritten by new bindings that have the same name. In other words, the last binding of a particular name to be applied takes effect. When you import an application, bindings are applied in the following order:

1. Application bindings generated by BizTalk Server that were not explicitly added to the application via a binding file, but that were explicitly selected by the user for export into the application .msi file.
 2. Binding files that have been explicitly added, and do not have a target deployment environment specified. Bindings in this set are applied in no specific order.
 3. Bindings that have been explicitly added, and that have an associated target deployment environment that matches the deployment environment selected for application import. Bindings in this set are applied in no specific order.
- **The host specified must exist.** To import an application from an .msi file, a host corresponding to the host specified in the application bindings contained in the .msi file must already exist in the BizTalk group or the import operation will fail. In addition, the host trust level must match.
 - **Dependencies can have significant effects upon import operations.** When you import an application that has a dependency on another application, the following rules apply:
 - If an application that you are importing depends on an artifact in another application, you must add a reference from the first application to the second application. The application and the required artifact must already exist in the destination group. The Import Wizard enables you to add the reference. However, if you are using the ImportApp command of BTSTask, you must add the reference to the application after import. For more information, see [How to Add a Reference to Another Application](http://go.microsoft.com/fwlink/?LinkId=155011) (<http://go.microsoft.com/fwlink/?LinkId=155011>). The Import Wizard matches the references to existing applications in the group and gives you the option to add a new

reference or change an existing reference. While BizTalk Server verifies that the referenced application exists, we recommend that you take the additional step of verifying that the referenced application contains the required artifact.

- When you install an application, you must also install any applications on which it depends. When you install an application that has a dependency on an artifact, such as a BizTalk assembly, which is contained in another application, you must first install the application that contains the artifact. For example, if you want to install Application A, and it depends on an assembly in Application B, you must install Application B first. Then you can install Application A. For more information about installing a BizTalk application, see [How to Install an Application](#).
- If you want to import an application into a different BizTalk group and run it in that group, you must also import any artifacts on which this application depends. You can do this by first importing an application that contains the referenced artifact, or by adding the needed artifact to the application that requires it. For more information about importing a BizTalk application, see [How to Import an Application from an .msi File](#).

For more considerations and information about importing a BizTalk application from an .msi file, see [How to Import a BizTalk Application](#) (<http://go.microsoft.com/fwlink/?LinkID=154827>).

How to Import an Application

For instructions on importing a BizTalk application from an .msi file, see [How to Import a BizTalk Application](#) (<http://go.microsoft.com/fwlink/?LinkID=154827>).

How to Install an Application

For information about installing an application, see [How to Install a BizTalk Application](#) (<http://go.microsoft.com/fwlink/?LinkID=154728>).

How to Import Bindings from a Binding File

You can import the bindings of a BizTalk application into another existing BizTalk application by using a binding file. You can also import all the bindings for a group into a group or the bindings of an assembly into an application.

- For information about various considerations to keep in mind when importing bindings, see [Importing Bindings](#) (<http://go.microsoft.com/fwlink/?LinkId=155012>).
- For instructions on importing bindings into a BizTalk application, see [How to Import Bindings into a BizTalk Application](#) (<http://go.microsoft.com/fwlink/?LinkId=155013>).

Testing an Application

For information about the steps involved in deploying the artifacts of a BizTalk application to a test environment for testing and debugging, see [Testing Tasks for BizTalk Application Deployment](#) (<http://go.microsoft.com/fwlink/?LinkID=154825>).

How to Add a Reference to an Application

This topic provides information about using the BizTalk Server Administration console to add a reference from one application to another. You add a reference when you want to use an artifact in the current application that already exists in another application in the same BizTalk group. If you want to use the EDI-specific schemas, pipelines, and orchestrations (deployed in the BizTalk EDI Application) from another application such as BizTalk Application 1, you must add a

reference to the BizTalk EDI Application. You should not add custom artifacts to the BizTalk EDI Application.

For information about:

- Adding a reference to another application, see [How to Add a Reference to Another Application](http://go.microsoft.com/fwlink/?LinkID=155011) (http://go.microsoft.com/fwlink/?LinkID=155011).
- Adding a reference to the BizTalk Server EDI application, see [How to Add a Reference to the BizTalk Server EDI Application](http://go.microsoft.com/fwlink/?LinkID=155026) (http://go.microsoft.com/fwlink/?LinkID=155026).

Updating an Application

This section describes how to update a BizTalk application after it has been deployed into a production environment.

In This Section

- [Best Practices for Updating Applications](#)
- [Known Issues with Updating Applications and Artifacts](#)
- [Updating Applications and Artifacts](#)

Best Practices for Updating Applications

This topic describes best practices that you should consider using when updating BizTalk applications and artifacts.

Versioning

Implement a versioning strategy

- A good versioning strategy is essential if long-running transactions are used or the BizTalk application cannot be taken down to do upgrades or bug fixes. You should plan the versioning strategy of all BizTalk artifacts: schemas, maps, custom adapters, pipelines, pipeline components, orchestrations, business rules, BAM, and custom classes called in orchestrations and maps.

Match the assemblies in the BizTalk Management database and the global assembly cache (GAC)

- Ensure that the same versions of assemblies are in the BizTalk Management database as in the GAC, so that your application functions properly. If you do not always install an assembly in the GAC when you deploy it, you might have different versions in the GAC and the BizTalk Management database, which will cause processing errors during run time.

Use the BizTalk Assembly Checker and Remote GAC tool to verify versioning

- The **BizTalk Assembly Checker and Remote GAC tool** (BTSAssemblyChecker.exe) checks the versions of assemblies deployed to the BizTalk Management database and verifies that they are correctly registered in the GAC on all BizTalk Server computers. You can use this tool to verify that all the assemblies containing the artifacts of a certain BizTalk application are installed on all BizTalk nodes. The tool is particularly useful in conjunction with a solid versioning strategy to verify that the correct version of a set of assemblies is installed on each BizTalk machine, especially when side-by-side deployment approach is used.

- The tool is available with the BizTalk Server 2010 installation media at Support\Tools\x86\BTSAsemblyChecker.exe.

Use a Versioning Product

- You should use a versioning product, such as Microsoft Visual Studio® Team Foundation Server 2010, for tracking and versioning BizTalk artifacts. For more information about Microsoft Visual Studio® Team Foundation Server 2010 see [Microsoft Visual Studio® Team Foundation Server 2010](http://go.microsoft.com/fwlink/?LinkId=210637) (http://go.microsoft.com/fwlink/?LinkId=210637)

Factor artifacts into multiple BizTalk applications

- In order to perform assembly versioning of BizTalk artifacts, your BizTalk solution assemblies need to be factored (packaged) in such a way to allow for BizTalk Server versioning. For more information about factoring, see [Adding Artifacts to an Application](#).

Updating an Application

Use an .msi file to update an application

- Upgrading applications is typically a deliberate and precise operation in production. When you upgrade an application, you should normally use a manual checklist. However, you may be able to streamline certain steps by using an .msi file. When you use an .msi file, you can wrap up your application artifacts into a distributable package. An .msi file is particularly useful when you roll out updated DLLs to multiple runtime boxes or perform a group-level deployment. When you create an .msi file, you should exclude all other unchanged resources and bindings from the package.
- If you update a BizTalk assembly, you should stop, unenlist, re-enlist, and then start BizTalk artifacts manually before and after you import and install the .msi file. For more information about updating a BizTalk assembly, see [Checklist: Updating an Assembly](#).
- If you upgrade a BizTalk Server assembly using side-by-side versioning, you will have to perform manual steps before and after using the .msi file. For more information about the manual steps that are required, see [Checklist: Updating an Application Using Side-by-Side Versioning](#).

Updating an Assembly

Increment the version of an assembly in a production environment

- If you are updating an assembly that is running in a production environment, you should always increment the assembly version number.

Update the GAC with an updated assembly

- When you update an assembly containing an orchestration, schema, or map, you must update the GAC with the assembly containing the new version. Otherwise, BizTalk Server will use the outdated version. To do this, on each computer that is running an instance of the host to which an application is bound, uninstall from the GAC the outdated version of the assembly containing the updated artifact and make sure that the new version is installed.

Restart a host instance after updating an assembly

- If a BizTalk assembly in an existing application is updated, you may need to restart host instances for the changes to take effect. Restarting a host instance stops all other applications that are running on the host instance.

Updating an Artifact

Undeploy a dependent artifact before the artifact that it depends on

- If you are undeploying an artifact on which another artifact depends, you must undeploy the dependent artifact first.

Note

If you do not undeploy the dependent artifact first, the BizTalk Server Administration console will display a warning and prevent you from undeploying artifacts in the incorrect order.

Do not stop an artifact that another application depends on

- If you stop an artifact in one application (which may result from stopping the entire application) that another application depends on, the dependent application will not function correctly. For more information about stopping an application, see [How to Start and Stop a BizTalk Application](http://go.microsoft.com/fwlink/?LinkID=154729) (<http://go.microsoft.com/fwlink/?LinkID=154729>).

Add a reference to an assembly before moving an artifact

- When you move an artifact to a new application, any other artifacts on which it has dependencies are also moved unless the new application has a reference to the application(s) containing the artifacts on which the moved artifact depends. Also, any artifacts that have dependencies on the moved artifact are moved unless the application(s) containing them have a reference to the new application. When moving an artifact, you are shown the list of other artifacts that will also be moved.

Updating Bindings

Automate the reconfiguration of bindings

- When you update an assembly in an application, its bindings are often overwritten or else the assembly may not be bound at all, forcing you to manually reconfigure bindings. You can automate this process by using a binding file. If you are updating the same version of an assembly, you can first export a binding file for the assembly, then update the assembly, then import the assembly into the application, and then reapply the previous bindings by importing the binding file. If you are updating an assembly with a newer version, you can export a binding file, edit the file to reflect the new assembly version, import the new assembly into the application, and then apply the new bindings by importing the binding file. For more information about binding files, see [How to Export Bindings to a Binding File](#). For more information about editing a binding file, see [Customizing Binding Files](#) (<http://go.microsoft.com/fwlink/?LinkID=155000>).

Starting or Stopping an Application

Stop an application to update artifacts

- If you do not stop an application to update artifacts in the application, you need to temporarily halt publication to the MessageBox database by disabling endpoints, and stop and unenlist any running instances. To stop and unenlist running instances, all dehydrated or suspended instances must be either manually resumed and completed, or terminated.
- Although it is not required to stop an application in order to update an artifact or install the application, we recommend that you always stop an application when you update an artifact.

See Also

[How to Export Bindings to a Binding File](#)

Known Issues with Updating Applications and Artifacts

Updating an Application

Removing or deleting an artifact does not remove it from all locations

Removing or deleting an artifact deletes it from the BizTalk Server databases, so that it no longer appears in the administration console or in the list of artifacts for an application generated by the BTSTask ListApp command. It does not remove the artifact from the Windows registry, the global assembly cache (GAC), a virtual directory, or the file system, if it exists in any of these locations. In the case of send ports, send port groups, receive ports, and receive locations, which exist only in the BizTalk Management database, this operation deletes the artifact entirely.

Updating an Artifact

Removing or changing the state of an artifact can cause an application not to work

When you add a reference from one application to another and make any changes to the state of an artifact on which another application depends, or you remove the artifact, the application that has the dependency will not function correctly. For more information about changing the state of an artifact, see the section on the appropriate artifact in [Managing Artifacts](http://go.microsoft.com/fwlink/?LinkID=154725) (<http://go.microsoft.com/fwlink/?LinkID=154725>).

.NET policy files are not supported

The use of .NET policy files is not supported in BizTalk Server 2010. This is because a policy file may not work as expected. Policy files redirect .NET to a specified assembly version in the GAC, but BizTalk Server often accesses assemblies and artifact data from a cache or the BizTalk Management database. Depending on the artifact type, caching situation, and whether host instances are restarted, the policy file may not do what is desired.

Updating an Assembly

Changes to an assembly may not take effect if the host is not stopped

BizTalk assemblies must follow .NET versioning rules. The main implication of this is that a BizTalk project, once built against a particular version of another .NET project or assembly (including BizTalk projects), continues to use that version until it has been rebuilt against a newer version.

A common problem occurs during development related to .NET versioning when the version numbers on a BizTalk project are not changed and the assembly is redeployed without stopping and starting the BizTalk host instance that the types are loaded into.

When the process is run again, the changes do not take effect. This is due to the way in which .NET assemblies are loaded into memory. Because the host already has an in-memory copy of the assembly, it does not reload the assembly when a new copy is put into the global assembly cache (GAC). For example, if version 1.0.0.0 of an assembly with an orchestration is deployed and running, and changes are made to the orchestration but the version number is not changed, then the changes do not take effect. After the host instance is stopped, the in-memory copy of the assembly is released and when the host instance starts again it reloads the new copy of the assembly and gets the changes. If a new version was deployed, for example version 2.0.0.0, and it was loaded, then the changes would take effect.

Changing the assembly version may break the relationship between an assembly and items that reference it by version

In .NET Framework development it is typical to update the assembly version number to the current build number when a build takes place. However, when developing a BizTalk solution, changing the assembly version number can break the relationship between an assembly and the dependent items that reference the DLL by its assembly version number. The following table lists items that refer to a BizTalk Server assembly by using its version number and the effect of changing an assembly version number.

Entity	Effect of changing assembly version number
Binding files	<p>Changing the assembly version number causes any existing binding files that reference the assembly to fail. This is because the binding file references the assembly by attributes including its version number.</p> <p>You can update the version information in the binding file by using Notepad or another editor. You can also redeploy the solution and then regenerate the binding file by using the BizTalk Server Administration console. Finally, you can use scripts to automate deployment and versioning. For more information about deployment, see Deploying and Managing BizTalk Applications (http://go.microsoft.com/fwlink/?LinkID=154210).</p>
BAM tracking profile definition (.btt) files	<p>Changing the assembly version number causes any existing BAM tracking profile definition files to fail. The BAM tracking files are in a binary file format so they cannot be edited and instead must be regenerated. If BAM tracking profiles are required it may be necessary to do either of the following:</p> <ul style="list-style-type: none"> • Avoid frequently updating version numbers during the build process • Delay building BAM tracking profiles until version numbers are stable
Web services published by using the Web Services Publishing Wizard	<p>When the Web Services Publishing Wizard is used to produce an ASP.NET Web interface, the assembly version of the BizTalk Server assembly is included in the ASP.NET source code. The assembly version number is used at</p>

Entity	Effect of changing assembly version number
	<p>runtime by the ASP.NET interface as part of the bodyTypeAssemblyQualifiedName property of the Web service operation. If the version number of the BizTalk Server assembly changes without updating the bodyTypeAssemblyQualifiedName property, then subsequent Web service operations will be rejected by BizTalk Server.</p> <p>If the receive location uses the XmlDefaultPipeline, the subscription relies on the document type. It uses the embedded assembly information and will fail if the assembly does not exist. If you use the PassThruPipeline (which is the default if you expose a port and let the wizard create the receive location), the subscription ignores this embedded assembly information.</p>

Updating Applications and Artifacts

This section describes how to update a running application.

In This Section

- [Implementing a Versioning Strategy](#)
- [Performing a Simple Update](#)
- [How to Update an Assembly](#)
- [Updating an Artifact](#)
- [Updating Using Side-by-Side Versioning](#)
- [Updating the Bindings in an Application](#)

Implementing a Versioning Strategy

Versioning is the act of updating the implementation of an artifact and incrementing its version number.

General Versioning Issues

BizTalk application versioning can become an issue when you need to run two versions of a BizTalk solution side-by-side, or if you cannot schedule BizTalk application downtime to deploy a new version. If you do not need to run two versions of the solution simultaneously (for example, if you have no long-running orchestrations), then it is perfectly acceptable to undeploy the old version and deploy the new version as a versioning strategy (no assembly versioning). This is a possible versioning strategy, although we still recommend incrementing the file version number (to let you know which version is deployed on the BizTalk servers). For more information about updating an application that is deployed, see [Checklist: Updating an Assembly](#).

If you need to support long-running orchestrations, and/or you need to perform BizTalk application deployments with no BizTalk application downtime, then a solid BizTalk versioning strategy needs to be implemented and practiced end-to-end for the different versioning scenarios. This includes .NET assembly versioning and versioning of all BizTalk artifacts. This includes schemas, maps, pipelines, pipeline components, orchestrations, custom adapters, custom classes called in orchestrations and maps, business rules, and BAM. For more information about side-by-side versioning, see [Updating Using Side-by-Side Versioning](#).

Versioning an Assembly

When you update an assembly, you have a choice between the following:

- Choosing a fixed assembly version for a given deliverable and incrementing only the file version number.
- Incrementing both the assembly version and the file version during the course of development.

These approaches are compared in the following table:

Fixed assembly version, dynamic file version	Dynamic assembly version, fixed or dynamic file version
Assembly version number = Fixed number File version number = Build number	Assembly version number = Build number File version number = Build number
BizTalk Server runtime may pick up the wrong version of the assembly if multiple assemblies are installed.	BizTalk Server always runs the latest version of the assembly, even if multiple assemblies are installed.
Only one version of the solution can be deployed at any time.	Different versions of the solution can be deployed side by side (although other aspects of the solution, such as schema definitions, may conflict).
BizTalk Host needs to be restarted to force the loading of updated assemblies.	Forces BizTalk Server to load new assemblies.
Requires less work to create a new deployment because files that reference the assembly version number (for example, binding files and tracking profiles) do not need to be edited.	Requires more work for deployment because files that reference the assembly version number need to be kept updated with the new version.

You may choose to use the fixed assembly version and dynamic file version approach if you are prototyping a system or developing any other type of project that will not be released. If you do not intend to deliver the application to an end user, you can streamline deployment tasks and reduce broken dependencies by fixing the assembly version and incrementing the file version number. For version tracking, you must remember to increment the file version number for each build.

If you are building a project that will be delivered to an end user, you should consider incrementing the assembly version number and, optionally, storing a meaningful file version number. While this approach incurs the added effort of modifying build numbers and associated dependencies, it ensures that the latest versions of your assemblies are used. By using automated deployment scripts, you can lessen the impact of versioning. To view deployment samples, see [Application Deployment \(BizTalk Server Samples Folder\)](http://go.microsoft.com/fwlink/?LinkId=155134) (<http://go.microsoft.com/fwlink/?LinkId=155134>) in BizTalk Server Help.



Note

You should choose the versioning mechanism that ensures that the proper files are delivered and that simplifies maintenance and enhancement.

For more information about versioning issues, see [BizTalk Server Project Versioning](http://go.microsoft.com/fwlink/?LinkId=154209) (<http://go.microsoft.com/fwlink/?LinkId=154209>) in BizTalk Server Help.

Performing a Simple Update

A simple update of a BizTalk application or artifact requires no downtime, requires no code changes, and can be performed by an IT Pro or business analyst singlehandedly. These types of changes are typically as simple as adding an additional send port to a send port group, adding an additional send port with appropriate subscription filters, changing the URI of a send port, or updating a rule in the BRE. For more information, see [Creating and Using Business Rules](http://go.microsoft.com/fwlink/?LinkId=155136) (<http://go.microsoft.com/fwlink/?LinkId=155136>) in BizTalk Server Help.

How to Update an Assembly

This topic describes how to update the version of an assembly and the application that an assembly is deployed to using Visual Studio 2010.



Note

If you are updating an assembly with a new version, you do not need to restart the application.

Prerequisites

To perform the procedures in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group.

Updating an Assembly

► To increment the version number of an assembly

1. In Solution Explorer, right-click the BizTalk project, and then click **Properties**.
2. In the **Project Designer**, click the **Application** tab.
3. In the right pane, click **Assembly Information**.
4. In the **Assembly Information** dialog box, specify values for the **Assembly Version** field to increase the assembly version number. You should increase only the major or minor version number. The major version number is the first digit in the sequence (*n*.0.0.0); the minor version number is the second digit in the sequence (0.*n*.0.0). BizTalk Server will not recognize a version number change that is later in the sequence, such as 0.0.*n*.0 or

0.0.0.*n*.

5. Click **OK** to close the **Assembly Information** dialog box.
6. Save the project.



Note

Use the Pipeline Designer and BizTalk Explorer Object Model to avoid schema collisions when incrementing assembly versions.

► **To change the application that an assembly is deployed to**

1. In Solution Explorer, right-click the BizTalk project, and then click **Properties**.
2. In the **Project Designer**, click the **Deployment** tab.
3. In the right pane, specify the application name in the **Application Name** field.
4. Ensure that the **Install to Global Assembly Cache** property is set to **True**.



Note

When you deploy the solution, the assemblies will be deployed into the new application and installed in the GAC.

Updating an Artifact

Dependencies between artifacts in two or more BizTalk applications can have a significant effect on application deployment and maintenance. An artifact is dependent on another when it needs to use that artifact in order to function properly, for example an orchestration that needs to use a specific pipeline in order to transmit messages correctly.

To update an artifact in an application, you must first undeploy it, along with any artifacts that depend on it. When artifacts that have dependencies exist in the same application, BizTalk Server automatically handles the undeployment and redeployment tasks for the updated and dependent artifacts. When artifacts that have dependencies exist in different applications, however, this is not the case. If you want to update an artifact in one application, and an artifact in another application has a dependency on it, you must undeploy and redeploy the dependent artifact manually as follows:

1. Stop, unenlist, and unbind the dependent artifact.
2. Update the artifact on which it depends.
3. Bind, enlist, and start the dependent artifact.

To avoid the need to perform manual steps to update an artifact on which other artifacts depend, you can try to keep all artifacts with dependencies together in the same application. This is not always possible, however, because most types of artifacts must be unique in a BizTalk group. You cannot have the same artifact in two different applications in the same group, even if both applications contain artifacts that depend on the same artifact. For more information about the issue of unique artifacts, see [Artifacts That Must Be Unique in an Application or Group](http://go.microsoft.com/fwlink/?LinkId=155141) (<http://go.microsoft.com/fwlink/?LinkId=155141>) in BizTalk Server Help.

You can add the needed artifact to one application and then add a reference to that application from any other applications containing artifacts that depend on it. When you add a reference to an application, artifacts in the application can use any artifacts in the application that it references. For instructions on adding a reference, see [How to Add a Reference to Another Application](http://go.microsoft.com/fwlink/?LinkID=155011) (http://go.microsoft.com/fwlink/?LinkID=155011) in BizTalk Server Help.

For a checklist of tasks for updating artifacts in a BizTalk application, see [Checklist: Update the Artifacts in a BizTalk Application](http://go.microsoft.com/fwlink/?LinkID=155647) (http://go.microsoft.com/fwlink/?LinkID=155647) in BizTalk Server Help.

See Also

[Updating Applications and Artifacts](#)

Updating Using Side-by-Side Versioning

If you are not able to schedule downtime, or have very long-running orchestration instances that cannot be terminated, side-by-side versioning may be required. In this type of upgrade, two versions of the same application or application artifacts run side-by-side. The .NET runtime inherently allows for same-named but differently versioned assemblies to be deployed and running, and BizTalk Server also allows it.

Side-by-side versioning of applications is useful when you want to roll out a major application upgrade incrementally, for example making it available to a subset of business partners initially, rather than to all partners at once. Using this approach allows you to continue running the existing application to service the users who are not yet using the new version until you are ready to completely cut over to the new version.

You do not create application versions in the same manner that you create assembly versions, by incrementing the version number. Instead, you create a new application that has a different name from the original application, and populate it with the new versions of the application artifacts.

Because many types of artifacts, such as assemblies, can exist in only one application in a BizTalk group, you must increment the version number of any assemblies that already exist in the group before you can deploy them into the new application.

For a step-by-step list of tasks required to update an application or orchestration using side-by-side versioning, see [Checklist: Updating an Application Using Side-by-Side Versioning](#) and [Checklist: Updating an Orchestration Using Side-by-Side Versioning](#). For detailed instructions on how to have side-by-side deployment of applications, see "[How to Deploy a New Version of an Application to Run Side-by-side with an Existing Version](#)" (http://go.microsoft.com/fwlink/?LinkID=155143) in BizTalk Server Help.

In This Section

- [How to Update a Map Using Side-by-Side Versioning](#)
- [How to Update a Pipeline Using Side-by-Side Versioning](#)
- [Updating a Schema Using Side-by-Side Versioning](#)

How to Update a Map Using Side-by-Side Versioning

Some BizTalk artifacts, such as maps, are chosen by fully-qualified strong name (FQSN), in which case the bindings include the version used. This allows two or more maps to coexist side

by side in the BizTalk Server Administration console. As a result, you can select one of the maps for inbound mapping in the receive location properties or outbound mapping in the send port properties.

Prerequisites

To perform the procedures in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group.

► To add a second map side by side to an existing map

1. Open Visual Studio, and then open the project containing the map.
2. Open the map in the assembly, and make a code change to the map.



Note

If you call a map from an orchestration, and the map reference is hard-coded, you may need to make code changes to the orchestration itself.

3. Change the version number of the assembly:
 - a. In Solution Explorer, right-click the BizTalk project, and then click **Properties**.
 - b. In the **Project Designer**, click the **Application** tab.
 - c. In the right pane, click **Assembly Information**.
 - d. In the **Assembly Information** dialog box, specify values for the **Assembly Version** field to change the assembly version number. You should change only the major or minor version number. The major version number is the first digit in the sequence (**n**.0.0.0); the minor version number is the second digit in the sequence (0.**n**.0.0).
 - e. Click **OK** to close the **Assembly Information** dialog box.
4. Compile the assembly.
5. Deploy the assembly to the group (and all computers).

Modifying a Map to Reflect Updated Version Numbers

.NET assemblies can be invoked from within a map by using the Scripting functoid. This provides a great deal of flexibility and enables the developer to solve many different custom mapping problems. It also imposes a unique constraint on the map—it must internally reference not only the assembly type name but also the full assembly version number being invoked. As a consequence, if the version number of the assembly called by the map changes, all of the links that reference the assembly will break.

To avoid this issue we recommend that if assemblies are required to be called from a map, a specific assembly is created to hold only map functionality and that the assembly version number of this assembly is fixed. In this way, other helper functions can have the assembly version updated without breaking the maps.

If an assembly referenced from a map is changed after map development then consider updating the map file outside of the Map Editor to reflect the updated version numbers.

► To modify a map file to reflect updated version numbers

1. Using the **Start** menu, open **Notepad**.
2. In **Notepad**, on the **File** menu, click **Open**. In the **Open** dialog box, select the map file you want to modify, and then click **Open**.
3. On the **Edit** menu, click **Find**. In the **Find** dialog box, enter **Assembly=**, and then click **Find Next**.
4. If there is a script reference to an external assembly, Notepad should find an XML element like the following:

```
<Script Language="ExternalAssembly"
Assembly="Contoso.Scripts, Version=2.0.0.0, Culture=neutral,
PublicKeyToken=
<token>
" Class="Contoso.Scripts" Function="CalculateValue"
AssemblyPath="Contoso.Scripts.dll"/>
```

5. Update the version number. If there are multiple instances, use **Replace** on the **Edit** menu.
6. Save the file.



Note

You can now open the map using the Map Editor.

How to Update a Pipeline Using Side-by-Side Versioning

The simple way to use a new pipeline added by side-by-side versioning is to select the newly deployed pipeline version in the send port or receive location. This will replace the old pipeline with the new one. However, if you need true side-by-side functionality for backwards-compatibility, then you must create new send ports and receive locations and bind them to the new pipeline version specified.

Prerequisites

To perform the procedure in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group.

► To add a new version of a pipeline component

1. In Visual Studio, create a new version of the pipeline component, and sign the assembly.
2. Add the pipeline component in the **Pipeline Components** folder (*<installation folder>\Pipeline Components*).
3. Add the pipeline component to your pipeline.
4. After building the pipeline or deploying your solution, remove the pipeline component from the **Pipeline Components** folder.
5. Add the pipeline component to the global assembly cache (GAC).

After you have completed these steps, the compiled pipeline assembly will refer to the correct version of the pipeline component, and the AppDomain used by BizTalk Server 2010 will find

the new version of the pipeline component in the GAC, rather than finding the previous version of the pipeline component in the Pipeline Components folder.

See Also

[Updating Using Side-by-Side Versioning](#)

Updating a Schema Using Side-by-Side Versioning

You can perform side-by-side versioning with schemas. You do so by adding a new version of the schema to an assembly, upgrading the version of the schema, while leaving the existing schema (and its version) unchanged.

If you increment a schema version, you must update the reference to the schema for any pipeline instances and pipeline components that use it. You must also update the maps that reference the schema (or create new maps) as well as any orchestrations that rely on the schema.

If you undeploy a schema, the previous version of the schema, if available, will become active.

Schema Resolution in Pipelines

If you add to an application an assembly that includes a new schema with the same message type as an existing schema in the BizTalk group, the schema with the latest version number will be used when schema resolution occurs in pipelines. If one message type refers to more than one .NET type, this ambiguity may cause pipeline execution failure. This is because schema lookup uses message type, the target namespace, and root name of the instance. This type of failure can occur for pipelines in any application that uses a schema with the same message type as the new schema. For more information about schema resolution, see [Schema Resolution in Pipeline Components](#) (<http://go.microsoft.com/fwlink/?LinkID=154207>) in BizTalk Server Help.

The schema resolution behavior for the XML Disassembler may require additional changes after adding a new version of a schema side by side with an older version. In certain cases, you may want to hard-code references in the pipeline disassembler properties in the Pipeline Designer to specific versions of the schemas. This enables you to avoid the dynamic resolution behavior, in which the XML Disassembler determines which schema to load by using the message type dynamically discovered at runtime from the message XML content.

Updating a Schema in an Orchestration

When you change the schema associated with multiple send and receive shapes in an orchestration, you can make changes much easier by setting the Message Type property for the message associated with each send or receive shape to Multi-part Message Type, rather than Schema. You can then set the Type property for the message part associated with each shapes to be the same schema. By so doing, you can subsequently change the schema by changing the Type property for each message part, rather than having to change the Message Type for each shape. For more information about making changes easier by this process, see the [8 Tips and Tricks for Better BizTalk Programming](#) white paper (<http://go.microsoft.com/fwlink/?LinkId=101594>).

Versioning Schemas

BizTalk Server takes a schema from the most recent version of the assembly containing it. This means that if you create a new version of a schema, the new version completely replaces all

previous versions of the schema. This works well when transactions are short-lived. However, transactions in the Business Process Management solution are long-lived: an order may take up to a year to complete.

To allow for the possibility of using multiple versions of a schema being in use, each schema in the solution includes a version number in its namespace. BizTalk pipelines determine the message type of a message based on the target namespace plus the root node name defined in the schema. For example, the namespace for the Order schema is as follows:

```
http://Microsoft.Samples.BizTalk.SouthridgeVideo.Schemas.Order.v1
```

Because the namespace identifies the schema, and inclusion of the version number makes the namespace unique to the schema, the new schema will be distinct from the older version. Thus, a new schema can be used without supplanting the old schema.

Changing the schema version can affect many parts of your solution, so this should be planned in advance. For more information about the effect of schema version changes, see [Schema Resolution in Pipeline Components](http://go.microsoft.com/fwlink/?LinkID=154207) (<http://go.microsoft.com/fwlink/?LinkID=154207>) in BizTalk Server Help.

When changing the version of schemas in an orchestration, use multipart message types. Doing so will result in greater flexibility when versioning schemas. For more information about the advantages of using multipart message types, see tip #1, “Always Use Multi-Part Message Types,” in the MSDN Magazine article [8 Tips And Tricks For Better BizTalk Programming](http://go.microsoft.com/fwlink/?LinkId=101594) (<http://go.microsoft.com/fwlink/?LinkId=101594>).

See Also

[Best Practices for Updating Applications](#)

Updating the Bindings in an Application

When you update an assembly in an application, its bindings are often overwritten or else the assembly may not be bound at all, forcing you to manually reconfigure bindings. To avoid this, you can use a binding file to update the same version of the assembly or update an assembly with a newer version.

Updating the Same Version of an Assembly

If the assembly has early bound ports or dynamic ports, and you changed the port configuration in the BizTalk Server Administration console, the settings will be lost when you update the assembly with an assembly having the same version number. You can export a binding file for the assembly that you are going to update. After updating the assembly, you can import the assembly into the application and then import its binding file to reapply the previous bindings.

Updating an Assembly with a Newer Version

You can update the version of an assembly by exporting the binding associated with a single assembly into a binding file, editing the binding file to reflect a new assembly version, and then importing the bindings of the assembly into the application. For more information about editing a binding file, see [Customizing Binding Files](http://go.microsoft.com/fwlink/?LinkID=155000) (<http://go.microsoft.com/fwlink/?LinkID=155000>) in BizTalk Server Help.

Managing Certificates

This section describes how to manage digital certificates used with BizTalk Server 2010. Its topics describe how to install certificates (and which folder to install them into), and how to configure BizTalk Server to use the certificate for MIME/SMIME and AS2, and for use with BizTalk adapters. Microsoft BizTalk Server 2010 can make use of public key infrastructure (PKI) digital certificates for purposes of document encryption and decryption, document signing and verification (non-repudiation), and party resolution.

For a checklist of steps to install the certificates, see [Checklist: Installing and Configuring Certificates](#).

In This Section

- [Best Practices for Managing Certificates](#)
- [Known Issues with Certificates in BizTalk Server](#)
- [Installing and Configuring Digital Certificates](#)

Best Practices for Managing Certificates

This section provides best practices for managing certificates in your Microsoft BizTalk Server 2010 environment.

Assess and Plan Your Use of Certificates

Do a Threat Model Analysis of your environment

- Do a Threat Model Analysis (TMA) of your environment to determine whether signing or encryption certificates will help you mitigate security threats.

Create a plan for public key certificates with partners

- Create a plan for sending public key certificates to and receiving public key certificates from partners. If you are not using signing certificates for party resolution, the public certificate can be attached to the message, in which case you do not need a copy of the certificate in your system beforehand.

Establish guidelines with partners for submitting public keys

- As part of your Service Level Agreement (SLA) with your partner, establish guidelines for submitting public keys, notifying you when their certificates are about to expire, and notifying you when they revoke a certificate.

Install Certificates

Download the certificate revocation list at set intervals

- Download the certificate revocation list (CRL) from your certification authority (CA) at set intervals. We recommend doing this once a week. The CRLs are downloaded automatically if there is a CA for the domain in which the BizTalk servers are joined.

Verify signing certificates

- Make sure you verify the signing certificates against the certificate revocation list. For more information about how to verify the signing certificates, see [How to Configure the](#)

[MIME/SMIME Decoder Pipeline Component](http://go.microsoft.com/fwlink/?LinkId=155145) (<http://go.microsoft.com/fwlink/?LinkId=155145>) in BizTalk Server Help.

Manage certificates with partners

- Make certificate management part of your partner management practices. When you add or remove a party from the BizTalk Server environment, we recommend that you add or remove the certificates associated with that partner.

Remove certificates before removing a host instance

- Before removing a host instance from a BizTalk server, remove the certificates in the personal store of the account under which the host instance is running.

Configure BizTalk Server to Use Certificates for MIME/SMIME

Avoid denial of service attacks for digital signatures

- Determine what you want to do with messages when BizTalk Server cannot validate the digital signature. Setting the Authentication property on the receive port will help prevent denial of service attacks.



Note

The Authentication - Drop messages and Authentication - Keep messages flags on the receive port require that the Party Resolution pipeline component be configured correctly, and that the parties are defined in BizTalk Server. For more information, see [Party Resolution Pipeline Component](http://go.microsoft.com/fwlink/?LinkId=155146) (<http://go.microsoft.com/fwlink/?LinkId=155146>) in BizTalk Server Help.

Create separate receive locations for encrypted and unencrypted messages

- If you plan to receive MIME-encrypted messages from some partners and unencrypted messages from other partners, create separate receive locations in different hosts for encrypted and unencrypted messages. When you expect only MIME-encrypted messages, configure the Allow Non MIME Message option in the Decode MIME/SMIME pipeline component to No.

Configure a BizTalk Adapter to Use Certificates

Test your connection to the target Web site

- If you are using SSL, ensure that you can connect to the target Web site with Microsoft Internet Explorer® before attempting to connect to the target Web site with the HTTP or SOAP transports. Verify that no dialog boxes are displayed in Internet Explorer when you connect to the target Web site. BizTalk Server has no mechanism for interfacing with any dialog boxes that might be displayed when connecting to the target web site. A dialog box may be displayed by Internet Explorer if the target Web site name does not match the name specified for the Web site in the SSL certificate or if the Root Certification Authority for the SSL certificate is not in the appropriate Trusted Root Certification Authorities store.

Use the SSL Diagnostics tool to analyze SSL connection issues

- The SSL Diagnostics tool is an optional component of the IIS Diagnostics Toolkit. You can download the IIS Diagnostics Toolkit from the [Internet Information Services Diagnostics Tools](http://go.microsoft.com/fwlink/?LinkID=64426) page (<http://go.microsoft.com/fwlink/?LinkID=64426>).

Exporting a Certificate from One BizTalk Group to Another

Ensure that an imported certificate is being used for its intended purpose

- If you import a certificate into a group, the imported certificate must have a usage property that is consistent with its intended use. To check the usage property, double-click the certificate in the **Certificates Management Console** interface, and then click the **Details** tab of the **Certificate** dialog box. Then click the **All** option for the **Show** drop-down list, and then click to select the **Key Usage** and/or **Enhanced Key Usage** fields to verify the intended purpose. If BizTalk Server attempts to use a certificate for other than its intended purpose a runtime error will occur.

Known Issues with Certificates in BizTalk Server

This section describes known issues with managing digital certificates used with BizTalk Server.

General Certificate Issues

Lack of connectivity to the certificate revocation list will cause a certificate to be rejected

This issue involves the following error: "There was an authentication error. The status of the certification authority (CA) that issued the certificate used to sign the message is unknown." This error can occur even when the signing certificate is valid when viewed under MMC (using the BizTalk Server user) on the BizTalk Server.

This condition can occur if the "Check Certificate Revocation" property is enabled on the S/MIME decoder component in the receive pipeline. When this property is set to true, BizTalk Server will try to query the Certificate Revocation List (CRL) to see if the incoming certificate has been revoked. It does not matter if the certificate itself is not revoked. If BizTalk Server cannot query the corresponding CRL because of connectivity issues, it will not accept the certificate. To troubleshoot this error, display the certificate's properties by double-clicking the certificate that you used. In its Details tab, you will see the attribute "CRL Distribution Point" in the field list. There should be several URLs in this attribute that point to the CRL on your CA server. The BizTalk server must be able to access any of these URLs to retrieve the CRL. Otherwise, the revocation checking will fail and the above error will be posted.

The Other People Certificate store is not initialized until accessed

This issue involves the following certificate store error when you try to add or modify send/receive ports/locations using the BizTalk Server Administrator console remotely: "Could not open certificate store." and "The system cannot find the file specified. (System)".



Note

You can modify these artifacts using the administration console if you log on directly to the BizTalk Server.

On a newly installed computer, the **Other People Certificate** store is not initialized unless you access it once. During the group configuration, you can initialize this **Other People Certificate** store, and as a result not see this error on a computer on which group configuration has been done.

BizTalk Server only supports one personal certificate for each BizTalk group

The personal certificate that is used by the BizTalk group is specified by setting the thumbprint of the personal certificate in the BizTalk group properties. A BizTalk group can represent an enterprise, a department, a hub, or another business unit.

AS2 Certificate Issues

The AS2 decoder will not validate that a certificate is configured on the host or for the destination party

The AS2 decoder will decrypt a message as long as the private certificate for that message is configured in the certificate store. However, the AS2 decoder will not validate that the decryption certificate is the same as the certificate configured on the host. The received message can be encrypted with more than one certificate.

BizTalk Server will be unable to decrypt a message saved in wire format if the certificate is not valid

Symptom

BizTalk Server is unable to decrypt an inbound AS2 message that was saved in wire format in the non-repudiation database.

Possible Cause

The certificate required to decrypt the message has expired or been revoked. This is more likely to occur if a certain period of time has elapsed since the AS2 message was saved in the non-repudiation database. If this occurs, you may not have immediate access to a valid certificate for the message.

Resolution

Acquire a valid certificate for decrypting the message.

If an AS2 message cannot be decrypted, the problem may be fixed by re-importing the certificate

Symptom

The AS2 Decoder encounters an exception when it attempts to decrypt an AS2 message, throwing the following error:

```
"The AS2 Decoder encountered an exception during processing. Details of the message and exception are as follows:
```

```
AS2-From:"PARTNER" AS2-To:"HOME" MessageID:"<137706.1178060412333@servername>"
```

```
MessageType: "unknown" Exception:"An error occurred when decrypting an AS2 message."
```

```
System.ArgumentNullException: Value cannot be null.
```

```
Parameter name: PayloadContentType
```

```
at Microsoft.BizTalk.Edi.Reporting.Common.Utilities.ValidateArgument(Object o,
```

```
String parameterName, Boolean isEmptyStringValidationRequired)
```

```
at Microsoft.BizTalk.EdiInt.Reporting.AS2MessageActivity.ValidateParameters()
```

```
at Microsoft.BizTalk.EdiInt.Reporting.AS2MessageActivity.Create() "
```

Possible Cause

The certificate used to decrypt the AS2 message needs to be reloaded into the Personal Store.

Resolution

Delete the existing certificate from the Personal Store, and then re-import the certificate into the Personal Store using the Certificate Import Wizard. Do so by right-clicking the **Certificate** folder under the **Personal Store**, pointing to **All Tasks**, and then clicking **Import**.

Use the same logon for the in-process host instance and the isolated host instance to ensure that personal store is recognized

The Personal certificate store will be available for message processing only if the user profile is loaded for the user whose logon credentials are associated with the host instance. The Personal store is used for signing and decryption certificates (the user's own private key). The user profile is loaded by default for the in-process host instance; however, the user profile is not loaded by default for the isolated host instance. You can have an application load the user profile for the isolated host. Alternatively, you can work around this issue by using the same logon for the in-process host instance and the isolated host instance.

Instead of having an application load the user profile, you can create an empty service to load the profile. For more information about creating an empty service, see [How to: Create Windows Services](http://go.microsoft.com/fwlink/?LinkId=155149) (<http://go.microsoft.com/fwlink/?LinkId=155149>) in Visual Studio Help.

After creating the empty service to load the profile, proceed as follows:

1. Click **Start**, and then click **Run** to open the **Run** dialog box.
2. In the **Run** dialog box, type **service.msc** and press **ENTER** to open the **Services** MMC snap-in.
3. Open the **Properties** dialog box for the service you created. Right-click the service and select **Properties**.
4. Click the **Log On** tab, select **This Account**, and then enter the logon name used for the isolated host instance.
5. Click **OK**.
6. Manually start the service to load the user profile for that logon user.

The Key Usage attribute of a certificate must match the certificate's use

Certificates used for AS2 transport must have the attributes required for their intended use. For signing and signature verification, the Key Usage attribute of the certificate must be Digital Signature. For encryption and decryption, the Key Usage attribute of the certificate must be Data Encipherment or Key Encipherment. You can verify the Key Usage attribute by double-clicking the certificate, clicking the **Details** tab in the **Certificate** dialog box, and checking the **Key Usage** field.

The certificate resolution list will be verified for an outgoing MDN if the AS2-To property is not set for the party

In the default agreement for an outgoing MDN, the certificate resolution list verification is performed. If you do not want this verification to be performed, verify that the correct AS2-To party property is set, so the receiving party can be resolved and the party properties can be determined. If so, the default agreement that prompts verification of the certificate resolution list

will not be used. You will also need to disable the Check Certification Revocation List property on the General page of the AS2 party properties.

Installing and Configuring Digital Certificates

This section provides procedures for managing digital certificates used with BizTalk Server.

In This Section

- [How to Install Certificates for BizTalk Server](#)
- [Configuring Certificates for MIME/SMIME Messages](#)
- [Configuring Certificates with Adapters](#)
- [How to Configure Certificates for Party Resolution](#)
- [How to Add Certificates to an Application](#)

How to Install Certificates for BizTalk Server

To help secure data transfer on BizTalk Server, you must add the appropriate certificate to the appropriate certificate store, and associate the certificates with the appropriate BizTalk artifacts. This topic describes how to display the Certificates Management Console interface for the Local Computer and Current User certificate stores, and how to install the appropriate certificate in the appropriate store.

The certificates shown in the following table are used to help sign, verify the signature for, encrypt, and decrypt messages.

Certificate Usage	Certificate Type	Certificate Store
Signature (outbound)	Own private key (.pfx)	Current User\ Personal store of each BizTalk server that hosts a MIME/SMIME encoder pipeline as each host instance service account
Signature verification (inbound)	Trading partner's public key (.cer)	Local computer\Other People store of each BizTalk server that hosts a MIME/SMIME decoder pipeline as each host instance service account
Encryption (outbound)	Trading partner's public key (.cer)	Local computer\Other People store of each BizTalk server that hosts a MIME/SMIME encoder pipeline
Decryption (inbound)	Own private key (.pfx)	Current User\Personal store of each BizTalk server that hosts a

Certificate Usage	Certificate Type	Certificate Store
		MIME/SMIME decoder pipeline as each host instance service account

Prerequisites

To perform the procedures in this topic you must be logged on as a member of the BizTalk Server Administrators group.

► To display the Certificates Management Console

1. Click **Start**, click **Run**, type **MMC**, and click **OK** to open the **Microsoft Management Console**.
2. Click the **File** menu, and then click **Add or Remove Snap-in** to display the **Add or Remove Snap-in** dialog box.
3. Click the **Add** button to display the **Add Standalone Snap-in** dialog box.
4. Select **Certificates** from the list of snap-ins, and then click **Add**.
5. Select **Computer account**, click **Next**, and then click **Finish**. This will add the **Certificates Management Console** interface for **Local Computer**.
6. Ensure that **Certificates** is still selected from the list of snap-ins, and then click **Add** again.
7. Select **My user account**, and then click **Finish**. This will add the **Certificates Management Console** interface for **Current User**.



Note

This displays the **Certificates Management Console** for the account that you are currently logged on as. If you need to import certificates into the Personal store for a service account then you should log on with the service account credentials first.

8. Click the **Close** button on the **Standalone Snap-in** dialog box.
9. Click the **OK** button on the **Add or Remove Snap-in** dialog box.

► To install a certificate for receiving encrypted messages

1. Request a private-public key pair for digital signatures from the certification authority (CA) for BizTalk Server to use.
2. Send to the partner the public key for encryption.
3. In BizTalk Server, log on as the service account for the host instance running the handler that will receive messages from the partner. Install the BizTalk Server private key certificate for decrypting messages in the personal store for the service account.



Note

For more information about the procedure used to install certificates for

encryption, see [How to Install the Certificates for Encrypted Messages](http://go.microsoft.com/fwlink/?LinkId=155156) (<http://go.microsoft.com/fwlink/?LinkId=155156>) in BizTalk Server Help. For more information about the tool used to import a certificate into the certificate store, see [Certificate Wizard Utility](http://go.microsoft.com/fwlink/?LinkId=155157) (<http://go.microsoft.com/fwlink/?LinkId=155157>) in BizTalk Server Help.

4. Have the partner install the BizTalk Server public key certificate for encrypting messages in the appropriate store. If the partner is using Windows 2000 Server, Windows Server 2003, or Windows Server 2008, they should install the public key in the Other People store.

To install a certificate for sending encrypted messages

1. Have the partner request a private-public key pair for encryption from the certification authority (CA).
2. Have the partner send you its public key for encrypting messages to be sent to the partner.
3. Have the partner install the private key certificate for decrypting messages in the appropriate store. If the partner is using Windows 2000 Server, Windows Server 2003, or Windows Server 2008, they should install the private key in the personal certificate store.



Note

For more information about the procedure used to install certificates for encryption, see [How to Install the Certificates for Encrypted Messages](http://go.microsoft.com/fwlink/?LinkId=155156) (<http://go.microsoft.com/fwlink/?LinkId=155156>) in BizTalk Server Help. For more information about the tool used to import a certificate into the certificate store, see [Certificate Wizard Utility](http://go.microsoft.com/fwlink/?LinkId=155157) (<http://go.microsoft.com/fwlink/?LinkId=155157>) in BizTalk Server Help.

4. In BizTalk Server, log on to the server that has a host instance running a handler that will send messages to the partner. Install the partner's public key certificate for encrypting messages sent to the partner in the Other People store.

To install a certificate for receiving signed messages

1. Have the partner request a private-public key pair for digital signatures from the certification authority (CA).
2. Have the partner send you its public key for digital signatures.
3. Have the partner install its private key certificate for signing messages in the appropriate store. If they are using Windows 2000 Server, Windows Server 2003, or Windows Server 2008, have them install the private key in the personal store for the account that will sign messages sent to BizTalk Server.



Note

For more information about the procedure used to install certificates for encryption, see [How to Install the Certificates for Encrypted Messages](http://go.microsoft.com/fwlink/?LinkId=155156) (<http://go.microsoft.com/fwlink/?LinkId=155156>) in BizTalk Server Help. For more

information about the tool used to import a certificate into the certificate store, see [Certificate Wizard Utility](http://go.microsoft.com/fwlink/?LinkId=155156) (http://go.microsoft.com/fwlink/?LinkId=155156) in BizTalk Server Help.

4. In BizTalk Server, log on to the server that has a host instance running a handler that will receive messages from the partner. Install the partner's public key certificate to verify their signature in the Other People store.

► To install a certificate for sending signed messages

1. Request a private-public key pair for digital signatures from the certification authority (CA) for BizTalk Server to use.
2. Send to the partner the public key for digital signatures.
3. In BizTalk Server, log on as the service account for the host instance running the handler that will send messages to the partner. Install the BizTalk Server private key certificate for signing messages in the personal store for the service account.



Note

For more information about the procedure used to install certificates for encryption, see [How to Install the Certificates for Encrypted Messages](http://go.microsoft.com/fwlink/?LinkId=155156) (http://go.microsoft.com/fwlink/?LinkId=155156) in BizTalk Server Help. For more information about the tool used to import a certificate into the certificate store, see [Certificate Wizard Utility](http://go.microsoft.com/fwlink/?LinkId=155157) (http://go.microsoft.com/fwlink/?LinkId=155157) in BizTalk Server Help.

4. Have the partner install the BizTalk Server public key certificate for verifying its digital signature in the appropriate store. If the partner is using Windows 2000 Server, Windows Server 2003, or Windows Server 2008, they should install the public key in the Other People store.

Configuring Certificates for MIME/SMIME Messages

To help secure data transfer on BizTalk Server, you must associate certificates installed in the certificate stores with the appropriate BizTalk artifacts. This applies to MIME/SMIME-encoded messages. It also applies to AS2 transport, which transports MIME/SMIME messages.

Use the following table as a reference for the certificate usage scenarios and configuration options available in BizTalk Server. Detailed procedures are provided in the topics listed in the "In This Section" heading at the end of this topic.

Certificate Usage	User Context	Certificate Store Location	Certificate Type	Configuration Parameters in the BizTalk Administration Console
Encryption	Account used by	Log on to each computer running BizTalk Server that will host S/MIME encoder	Trading	<ul style="list-style-type: none"> Specify values for the

Certificate Usage	User Context	Certificate Store Location	Certificate Type	Configuration Parameters in the BizTalk Administration Console
(Sending)	the host instance associated with the send handler	pipelines and import the encryption certificate into the Local Computer \ Other People store.	partner public certificate	<p>encryption certificate Common Name and Thumbprint on the Certificate page of the Send Port Properties dialog box.</p> <ul style="list-style-type: none"> Specify pipeline Encode options in the Configure Pipeline dialog box. The Configure Pipeline dialog box is displayed by clicking the button next to the Send pipeline drop-down list on the General page of the Send Port Properties dialog box.
Decryption (Receiving)	Account used by the host instance associated with the receive handler	<p>Log on to each computer running BizTalk Server that will host S/MIME decoder pipelines as each host instance service account, and import the decryption certificate to the Current User \ Personal store.</p> <p> Note For pipeline decryption to succeed on computers running IIS 6.0 or later, ensure that the account for the IIS application pool and the</p>	Own private certificate	<ul style="list-style-type: none"> Specify values for the decryption certificate Common Name and Thumbprint on the Certificates page of each Host Properties dialog box. Specify pipeline

Certificate Usage	User Context	Certificate Store Location	Certificate Type	Configuration Parameters in the BizTalk Administration Console
		<p>account used by the host instance associated with the receive handler are the same and that this account is a member of the <i><machineName></i>\IIS_WPG group. For more information about setting IIS process identity for IIS see Guidelines for Resolving IIS Permissions Problems (http://go.microsoft.com/fwlink/?LinkId=155161) in BizTalk Server Help. These processes must run under the same account to ensure that the account profile is loaded which in turns loads the registry keys required to perform decryption in the pipeline. For performance reasons, IIS does not load the account profile when starting the associated w3wp.exe process so the BizTalk Server host instance must be configured with the same account so that BizTalk Server will load the account profile and registry keys.</p>		<p>Decode options in the Configure Pipeline dialog box. The Configure Pipeline dialog box is displayed by clicking the button next to the Receive pipeline drop-down list on the General page of the Receive Location Properties dialog box.</p>
Signature (Sending)	Account used by the host instance associated with the send handler	Log on to each computer running BizTalk Server that will host S/MIME encoder pipelines as each host instance service account, and import the signature certificate to the Current User \ Personal store.	Own private certificate	<ul style="list-style-type: none"> Specify values for the signature certificate Common Name and Thumbprint on the Certificate page of the BizTalk Group Properties dialog box. <p> Note Only</p>

Certificate Usage	User Context	Certificate Store Location	Certificate Type	Configuration Parameters in the BizTalk Administration Console
				<p>one signature certificate can be specified per each BizTalk Server 2010 group.</p> <ul style="list-style-type: none"> Specify pipeline Encode options in the Configure Pipeline dialog box. The Configure Pipeline dialog box is displayed by clicking the button next to the Send pipeline drop-down list on the General page of the Send Port Properties dialog box.
Signature Verification (Receiving)	Account used by the host instance associated with the receive	Log on to each computer running BizTalk Server that will host S/MIME decoder pipelines and import the signature certificate to the Local Computer \ Other People store.	Trading partner public certificate	<ul style="list-style-type: none"> Specify values for the verification certificate Common Name and Thumbprint on the Certificates page of each

Certificate Usage	User Context	Certificate Store Location	Certificate Type	Configuration Parameters in the BizTalk Administration Console
	handler			<p>Party Properties dialog box.</p> <ul style="list-style-type: none"> Specify pipeline Decode options in the Configure Pipeline dialog box. The Configure Pipeline dialog box is displayed by clicking the button next to the Receive pipeline drop-down list on the General page of the Receive Location Properties dialog box. <p> Note The certificate used to verify a signature for a party must be unique from the certificates used to verify signature</p>

Certificate Usage	User Context	Certificate Store Location	Certificate Type	Configuration Parameters in the BizTalk Administration Console
				<p>es for other parties.</p> <p> Note Configuration of the Decode option requires that a pipeline with the MIME/S MIME decoder component is deployed.</p>
Party Resolution (Receiving)	Account used by the host instance associated with the receive handler	Log on to the BizTalk Server computer from which party resolution is being configured, and import the certificate into the Local Computer \ Other People store.	Trading partner public certificate	<ul style="list-style-type: none"> Specify values for the certificate Common Name and Thumbprint on the Certificates page of each Host Properties dialog box. Specify ResolveParty options in the Configure Pipeline dialog box. The Configure Pipeline dialog

Certificate Usage	User Context	Certificate Store Location	Certificate Type	Configuration Parameters in the BizTalk Administration Console
				<p>box is displayed by clicking the button next to the Receive pipeline drop-down list on the General page of the Receive Location Properties dialog box.</p> <p> Note Configuration of this option requires the use of a pipeline that contains the Party resolution component. The XMLReceive pipeline contains the Party resolution component.</p>

Certificate Usage	User Context	Certificate Store Location	Certificate Type	Configuration Parameters in the BizTalk Administration Console
HTTPS (Sending)	Account used by the host instance associated with the send handler	<p>SSL communication does not require a client certificate. Whether a client certificate is required is at the discretion of the destination Web server administrator. If the destination Web server requires a client certificate then follow these steps:</p> <ul style="list-style-type: none"> • Obtain the public certificate from the trading partner. • Log on to each computer running BizTalk Server as the account used by the host instance associated with the send handler. • Import the certificate into the Current User \ Personal store. <p>For information about configuring IIS to use SSL, see the Knowledge Base article HOW TO: Install Imported Certificates on a Web Server in Windows Server 2003 (http://go.microsoft.com/fwlink/?LinkId=155162).</p> <p>For information about how to obtain a certificate using Windows Server 2003 Certificate Services Web pages, see Use Windows Server 2003 Certificate Services Web Pages (http://go.microsoft.com/fwlink/?LinkID=69975).</p> <p> Note To use the Certificate Services Web page to obtain certificates from a Windows Server 2008 computer, see the Microsoft Knowledge Base article 922706 at http://go.microsoft.com/fwlink/?LinkId=155317 (http://go.microsoft.com/fwlink/?LinkId=155317).</p>	Trading partner public certificate	<ul style="list-style-type: none"> • HTTP Transport - Set the SSL client certificate thumbprint option on the Authentication tab of the HTTP Transport Properties dialog box. The HTTP Transport Properties dialog box is displayed by clicking the Configure button on the General page of the Send Port Properties dialog box. • SOAP Transport - Set the Client certificate thumbprint option on the General tab of the SOAP Transport Properties dialog box. The SOAP Transport Properties dialog box is displayed by clicking the Configure

Certificate Usage	User Context	Certificate Store Location	Certificate Type	Configuration Parameters in the BizTalk Administration Console
				button on the General page of the Send Port Properties dialog box.

In This Section

- [How to Configure BizTalk Server to Receive Encrypted MIME/SMIME Messages](#)
- [How to Configure BizTalk Server to Send Encrypted MIME/SMIME Messages](#)
- [How to Configure BizTalk Server to Receive Signed MIME/SMIME Messages](#)
- [How to Configure BizTalk Server to Send Signed MIME/SMIME Messages](#)

How to Configure BizTalk Server to Receive Encrypted MIME/SMIME Messages

This topic describes how to configure BizTalk Server to use certificates to receive encrypted MIME/SMIME messages. The procedure below also applies to configuring the receiving of encrypted messages over AS2 transport.

Prerequisites

To perform the procedure in this topic, you must be logged on as a member of the BizTalk Server Administrators group.

► To configure BizTalk Server to receive encrypted messages

1. Create a pipeline to receive encrypted messages, as follows:



Note

This step is not necessary when configuring AS2 transport for receiving encrypted messages because the AS2Receive and AS2EdiReceive pipelines that are included in BizTalk Server 2010 serve this function.



Note

The MIME/SMIME Decoder pipeline component performs both decryption and digital signature validation (when configured to perform both functions). Therefore, if you are configuring BizTalk Server to receive encrypted and signed messages, you can use the same receive pipeline. In other words, you do not have to create separate pipelines for decryption and digital signature validation.

- a. Create a receive pipeline and then drag the MIME/SMIME Decoder pipeline component into the Decode stage of the pipeline.
- b. In the **Properties** window, configure the MIME/SMIME Decoder pipeline component properties.

**Note**

Configuring the MIME/SMIME Decoder pipeline component properties includes setting the Check Revocation List property to True if you want to check the certificate revocation list for the certificates that senders use for signing messages that are being sent to BizTalk Server. Disabling this option increases the performance of the component. The certificate revocation list associated with a certificate is downloaded from the appropriate certificate services Web site. If the BizTalk Server cannot connect to the remote Web site, the message fails in the pipeline.

**Note**

You can configure pipeline properties for a receive location after the pipeline has been deployed into a BizTalk group using the BizTalk Server Administration console. You can configure different pipeline properties for each receive location in the BizTalk group.

- c. Build and deploy the receive pipeline.
2. Configure the receive location for receiving encrypted messages, as follows:
 - a. Add the BizTalk assembly that you created containing the receive pipeline to the BizTalk application including the receive locations to receive encrypted messages.

**Note**

This step is not necessary when configuring AS2 transport for receiving encrypted messages because the AS2Receive and AS2EdiReceive pipelines are included in the BizTalk EDI Application in BizTalk Server.

- b. Configure the receive locations in the BizTalk application with the receive pipeline that you created in the previous step.
3. Configure the host used as the receive handler for the receive location with the decryption certificate, as follows:
 - a. In the BizTalk Server Administration console, right-click a BizTalk host that is the handler for receiving the encrypted messages, and then click **Properties**.

**Note**

This procedure does not apply for AS2 transport available with BizTalk Server. It applies for the BizTalk MIME Decoder, not the AS2 Decoder. The AS2 Decoder will determine the certificate based on certificate information in the message.

- b. In the **Host Properties** dialog box, click **Certificate**, and then click **Browse**.

**Note**

The above procedure enables you to configure your BizTalk Server environment so that only certain hosts can receive and process particular messages. When you configure a host with the thumbprint of a certificate to use for message decoding and decryption, BizTalk Server creates an

application property for that host in the MessageBox database. Secure messages that are received and decrypted using this thumbprint are then only routed to this and other hosts that are configured with this thumbprint.

- c. In the **Select Certificate** dialog box, select the decryption certificate that you installed, and then close all the dialog boxes.

See Also

[Configuring Certificates for MIME/SMIME Messages](#)

How to Configure BizTalk Server to Send Encrypted MIME/SMIME Messages

This topic describes how to configure BizTalk Server to use certificates to send encrypted MIME/SMIME messages. The procedure below also applies to configuring the sending of encrypted messages over AS2 transport.

Prerequisites

To perform the procedure in this topic, you must be logged on as a member of the BizTalk Server Administrators group.

▶ To configure BizTalk Server to send encrypted messages

1. Create a pipeline to send encrypted messages, as follows:

Note

This step is not necessary when configuring AS2 transport for sending encrypted messages because the AS2Send and AS2EdiSend pipelines that are included in BizTalk Server serve this function.

- a. Create a send pipeline and then drag the MIME/SMIME Encoder pipeline component into the Encode stage of the pipeline.
- b. In the **Properties** window, configure the MIME/SMIME Encoder pipeline component Enable encryption property to **True**.

Note

You can configure the send pipeline component properties using the BizTalk Server Administration console after the pipeline has been deployed into a BizTalk group.

- c. Build and deploy the send pipeline.
2. Configure the send port for sending encrypted messages, as follows:
 - a. Add the BizTalk assembly that you created containing the send pipeline to the BizTalk application including the receive locations to receive encrypted messages.

Note

This step is not necessary when configuring AS2 transport for sending encrypted messages because the AS2Send and AS2EdiSend pipelines are included in the BizTalk EDI Application.

- b. Configure the send port in the BizTalk Application with the send pipeline that you

created in the previous procedure.

- c. Assign the encryption certificate that you installed by right-clicking the send port, clicking **Properties**, and then clicking **Certificate**. Click **Browse**, browse to the certificate that you want to assign to this send port, and then click **OK**.



Note

If the certificate does not exist on the local computer, in the **Thumbprint** box, type or paste the certificate thumbprint, and then click **OK**. The certificate thumbprint has the format HHHH HHHH, where H is a hexadecimal digit.

How to Configure BizTalk Server to Receive Signed MIME/SMIME Messages

This topic describes how to configure BizTalk Server to use certificates to receive signed MIME/SMIME messages. The procedure below also applies to configuring the receiving of signed messages over AS2 transport.

Prerequisites

To perform the procedure in this topic, you must be logged on as a member of the BizTalk Server Administrators group.

► To configure BizTalk Server to receive signed messages

1. Create a pipeline to receive signed messages, as follows:



Note

This step is not necessary when configuring AS2 transport for receiving signed messages because the AS2Receive and AS2EdiReceive pipelines that are included in BizTalk Server serve this function.

- a. Create a receive pipeline and then drag the MIME/SMIME Decoder pipeline component into the Decode stage of the receive pipeline.
- b. In the **Properties** window, configure the MIME/SMIME Decoder pipeline component properties.



Note

You can configure pipeline properties for a receive location after the pipeline has been deployed into a BizTalk group using the BizTalk Server Administration console. You can configure different pipeline properties for each receive location in the BizTalk group.

- c. Build and deploy the receive pipeline.
2. Configure a receive location for receiving signed messages, as follows:
 - a. Add the BizTalk assembly that you created containing the receive pipeline to the BizTalk application including the receive locations to receive signed messages.



Note

This step is not necessary when configuring AS2 transport for receiving signed messages because the AS2Receive and AS2EdiReceive pipelines are included in the BizTalk EDI Application in BizTalk Server.

- b. Configure the receive locations in the BizTalk application with the receive pipeline that you created in previous procedure.
3. Configure the party with a certificate for receiving signed messages, as follows:
 - Open the **Party Properties** dialog box in the BizTalk Server Administration Console, click the **Certificate** tab, click **Browse**, select the appropriate certificate, and then click **OK**.



Note

The certificate used to verify a signature for a party must be unique from the certificates used to verify signatures for other parties.

See Also

[Configuring Certificates for MIME/SMIME Messages](#)

How to Configure BizTalk Server to Send Signed MIME/SMIME Messages

This topic describes how to configure BizTalk Server to use certificates to send signed MIME/SMIME messages. The procedure below also applies to configuring the sending of signed messages over AS2 transport.

Prerequisites

To perform the procedure in this topic, you must be logged on as a member of the BizTalk Server Administrators group.

▶ To configure BizTalk Server to send signed messages

1. Create a pipeline to send signed messages, as follows:



Note

This step is not necessary when configuring AS2 transport for sending signed messages because the AS2Send and AS2EdiSend pipelines that are included in BizTalk Server serve this function.

- a. Create a send pipeline and then drag the MIME/SMIME Encoder pipeline component into the Encode stage of the pipeline.
- b. In the **Properties** window, configure the MIME/SMIME Encoder pipeline component Signature type property to ClearSign or BlobSign.



Note

If you are also using encryption, you can only select BlobSign.



Note

You can configure the send pipeline component properties using the BizTalk Server Administration console after the pipeline has been deployed into a BizTalk group.

**Note**

The MIME/SMIME Encoder pipeline component performs both encryption and digital signing (when configured to perform both functions). Therefore, if you are configuring BizTalk Server to send encrypted and signed messages, you can use the same send pipeline. In other words, you do not have to create separate pipelines for encryption and digital signing.

- c. Build and deploy the send pipeline.
2. Configure the send port for sending signed messages, as follows:
 - a. Add the BizTalk assembly that you created containing the send pipeline to the BizTalk application that includes the send ports to send signed messages.

**Note**

This step is not necessary when configuring AS2 transport for sending signed messages because the AS2Send and AS2EdiSend pipelines are included in the BizTalk EDI Application in BizTalk Server.

- b. Configure the send port in the BizTalk application with the send pipeline that you created in the previous procedure.
3. Configure the group with a certificate for sending signed messages, as follows:
 - a. Configure the BizTalk group with the signing certificate that you installed by expanding the BizTalk group in the BizTalk Server Administration Console, right-clicking **BizTalk Group**, and then clicking **Properties**.
 - b. Click the Certificate tab, click **Browse**, select the appropriate certificate, and then click **OK**.

See Also

[Configuring Certificates for MIME/SMIME Messages](#)

Configuring Certificates with Adapters

The following sections describe how to configure certificates for BizTalk Server adapters.

In This Section

- [How to Configure Certificates with an HTTP Adapter](#)
- [How to Configure Certificates with an MSMQ Adapter](#)
- [How to Configure Certificates with a SOAP Adapter](#)
- [How to Install and Configure Certificates with a WCF Adapter](#)

How to Configure Certificates with an HTTP Adapter

The HTTP send adapter can help secure a connection with servers that accept or require client certificates. If a client certificate is specified, the HTTP send adapter uses the certificate when connecting with servers that require or accept client certificates. If the client certificate is not specified and the destination server requires client certificates, the sender is not authenticated and the HTTP send adapter fails to send the message and follows the standard retry logic.

The HTTP send adapter uses the client certificate from the personal store of the account under which the BizTalk Server process is running. The certificate is specified by its thumbprint. If the HTTP send adapter fails to load the certificate for any reason, the message that it was sending is suspended.

When using an HTTP adapter to send an encrypted or signed message, configure the SSL certificate thumbprint HTTP transport property for the send port. In this property, specify the thumbprint of the certificate to use for message authentication.

Prerequisites

To perform the procedure in this topic, you must be logged on as a member of the BizTalk Server Administrators group.

► To configure BizTalk Server to send messages over an HTTP connection

1. In the BizTalk Server Administration console, create a new HTTP send port or open the properties for an existing HTTP send port.
2. Click **Configure** in the Transport section of the **Send Port Properties** dialog box.
3. Click **Authentication** in the **HTTP Transport Properties** dialog box.
4. In **Authentication type**, select **Anonymous**, **Basic**, **Digest**, or **Kerberos**.
5. If the authentication type is **Basic** or **Digest**, either select **Do not use Single Sign-On** (in which case you must specify the user name and password) or select **Use Single Sign-On** (in which case you must select the Affiliate Application).
6. In **SSL client certificate thumbprint**, enter the appropriate thumbprint.
7. Click **OK**, and then click **OK** again.

How to Configure Certificates with an MSMQ Adapter

The MSMQ send adapter can help secure a connection with servers that accept or require client certificates. If a client certificate is specified, the MSMQ send adapter uses the certificate when connecting with servers that require or accept client certificates. If the client certificate is not specified and the destination server requires client certificates, the sender is not authenticated and the MSMQ send adapter fails to send the message and follows the standard retry logic.

The MSMQ send adapter uses the client certificate from the personal store of the account under which the BizTalk Server process is running. The certificate is specified by its thumbprint. If the MSMQ send adapter fails to load the certificate for any reason, the message that it was sending is suspended.

When using an MSMQ adapter to send an encrypted or signed message, configure the Certificate Thumbprint MSMQ transport property for the send port. In this property, specify the thumbprint of the certificate to use for message authentication. Use this property in combination with the Use Authentication property to verify the message. Use the User Name and Password properties to gain access to queues.

Prerequisites

To perform the procedure in this topic, you must be logged on as a member of the BizTalk Server Administrators group.

▶ **To configure BizTalk Server to send messages over an MSMQ connection**

1. In the BizTalk Server Administration console, create a new MSMQ send port or open the properties for an existing MSMQ send port.
2. Click **Configure** in the **Transport** section of the **Send Port Properties** dialog box.
3. In the **MSMQ Transport Properties** dialog box, for **Authentication**, select **True**.
4. For **Certificate thumbprint**, enter the appropriate thumbprint.
5. Click **OK**, and then click **OK** again.

How to Configure Certificates with a SOAP Adapter

The SOAP send adapter can help secure a connection with servers that accept or require client certificates. If you specify a client certificate, the SOAP send adapter uses the certificate when connecting with servers that require or accept client certificates. If you do not specify a client certificate and the destination server requires client certificates, the sender is not authenticated and the SOAP send adapter fails to send the message and follows the standard retry logic.

The SOAP send adapter uses the client certificate from the Personal store of the account under which the BizTalk Server process is running. The SOAP adapter specifies the certificate by its thumbprint. If the SOAP send adapter fails to load the certificate for any reason, the message that it was sending is suspended.

When using a SOAP adapter to send an encrypted or signed message, configure the Client Certificate Thumbprint SOAP transport property for the send port.

Prerequisites

To perform the procedure in this topic, you must be logged on as a member of the BizTalk Server Administrators group.

▶ **To configure BizTalk Server to send messages over a SOAP connection**

1. In BizTalk Server Administration console, create a new SOAP send port or open the properties for an existing SOAP send port.
2. Click **Configure** in the **Transport** section of the **Send Port Properties** dialog box.
3. On the **General** tab, in **Authentication type**, select **Anonymous**, **Basic**, **Digest**, or **NTLM**.
4. If the authentication type is **Basic** or **Digest**, either select **Do not use Single Sign-On** (in which case you must specify the user name and password) or select **Use Single Sign-On** (in which case you must select the Affiliate Application).
5. In **SSL client certificate thumbprint**, enter the appropriate thumbprint.
6. Click **OK**, and then click **OK** again.

How to Install and Configure Certificates with a WCF Adapter

For information about installing and configuring certificates with WCF adapters, see [Installing Certificates for the WCF Adapters](http://go.microsoft.com/fwlink/?LinkId=155204) (http://go.microsoft.com/fwlink/?LinkId=155204) in BizTalk Server Help.

See Also

[How to Configure a WCF-BasicHttp Send Port](#)

[How to Configure a WCF-BasicHttp Receive Location](#)

[How to Configure a WCF-NetMsmq Send Port](#)

[How to Configure a WCF-NetMsmq Receive Location](#)

[How to Configure a WCF-NetTcp Send Port](#)

[How to Configure a WCF-NetTcp Receive Location](#)

[How to Configure a WCF-WSHttp Send Port](#)

[How to Configure a WCF-WSHttp Receive Location](#)

How to Configure Certificates for Party Resolution

When you use BizTalk Explorer to configure a party, you can configure the party so that the party is resolved by using its digital signature. If you configure the party this way, when BizTalk Server receives a message, it will use the public key certificate to determine who sent the message, and to resolve the sender to a known party in the BizTalk Server environment.

Prerequisites

To perform the procedure in this topic, you must be logged on as a member of the BizTalk Server Administrators group.

To configure party resolution to use a certificate

1. Open the **Party Properties** dialog box.
2. Click the **Certificate** tab, and then click **Browse**.
3. Select a certificate.
4. Click **OK**.

How to Add Certificates to an Application

For information about adding certificates to an application see [How to Add a Certificate to an Application](http://go.microsoft.com/fwlink/?LinkId=155217) (http://go.microsoft.com/fwlink/?LinkId=155217) in BizTalk Server Help.

Moving Databases

The recommended method for moving BizTalk Server databases (except for the Business Activity Monitoring (BAM) databases) is to configure BizTalk Server log shipping as described in the section [Disaster Recovery](#). With the exception of the databases used by BAM, all of the BizTalk Server databases can be backed up by using the **Backup BizTalk Server** SQL Server Agent job.

For more information about this job, see [How to Schedule the Backup BizTalk Server Job](http://go.microsoft.com/fwlink/?LinkId=154674) (http://go.microsoft.com/fwlink/?LinkId=154674) in BizTalk Server Help.

This section describes how to move the BAM-related databases and how to move the remaining BizTalk Server databases without first configuring BizTalk Server log shipping. This approach may be useful when upgrading the SQL Server computers that house the BizTalk Server databases or in other scenarios that are not related to disaster recovery.

In This Section

- [Moving BAM Databases](#)
- [Moving Non-BAM Databases](#)

See Also

[Managing BizTalk Server](#)

Moving BAM Databases

This section describes how to move the BAM-related BizTalk Server databases from one SQL Server computer to another.

In This Section

- [How to Move the BAM Primary Import Database](#)
- [How to Move the BAM Archive Database](#)
- [How to Move the BAM Star Schema Database](#)
- [How to Move the BAM Analysis Database](#)
- [How to Move the BAM Notification Services Databases](#)

See Also

[Moving Databases](#)

How to Move the BAM Primary Import Database

You can use this procedure to move the BAM Primary Import database to another server. From an end-to-end scenario perspective, moving the BAM Primary Import database involves two major steps:

- [Moving the BAM Primary Import Database](#)
- [Updating References to the New BAM Primary Import Database](#)

Prerequisites

You must be logged on with an account that is a member of the SQL Server sysadmin fixed server role to perform this procedure.

Moving the BAM Primary Import Database

Perform the steps in the following procedure to move the BAM Primary Import database.

► To move the BAM Primary Import database

1. Stop any BAM cube update and data maintenance SSIS packages, or prevent them from running until you have restored the BAM Primary Import database.
2. Stop all BizTalk Server services. For more information, see the topic [How To Start, Stop, Pause, Resume, or Restart BizTalk Server Services](http://go.microsoft.com/fwlink/?LinkId=154394) (<http://go.microsoft.com/fwlink/?LinkId=154394>) in BizTalk Server Help.
3. Stop the IIS service.
4. Stop the BAM Alerts Notification service:
 - a. Click **Start**, click **Run**, type **cmd**, and then click **OK**.
 - b. At the command prompt, type:
Net stop NS\$BamAlerts
5. Back up the BAM Primary import database on the old server. For instructions on backing up a database, follow the instructions on how to back up a database at [How to: Back Up a Database \(SQL Server Management Studio\)](http://go.microsoft.com/fwlink/?LinkId=156510) (<http://go.microsoft.com/fwlink/?LinkId=156510>) in SQL Server Books Online.
6. Copy the BAM Primary Import database to the new SQL Server computer.
7. Restore the BAM Primary import database on the new server. For instructions on restoring the database, follow the instructions on how to restore a database at [How to: Restore a Database Backup \(SQL Server Management Studio\)](http://go.microsoft.com/fwlink/?LinkId=156511) (<http://go.microsoft.com/fwlink/?LinkId=156511>) in SQL Server Books Online.



Note

If you restore the BAM Primary Import database from a backup, then you should also restore the BAM Archive, BAM Star Schema, and BAM Analysis databases by using a backup older than the BAM Primary backup.

Updating References to the New BAM Primary Import Database

After you have moved the database, you must update all the references to the new BAM Primary Import Database. The following references must be updated:

- Update all the BizTalk databases with the new server name. You can do so by using the UpdateDatabase.vbs script. See [To update BizTalk Databases with the new server name](#).
- Update the Web.config file for the BAM portal. See [To update the Web.config file for the BAM portal](#).
- Update the reference to the BAM Primary Import Database in all BAM Livedata Microsoft Excel files. See [To update reference in BAM Livedata Microsoft Excel files](#).
- Update the new server and database names in all BAM analysis SSIS packages. See [To update server and database names in all BAM SSIS packages](#).
- Update the new server and database names in data sources for all OLAP cubes. See [To update server and database names in data sources for all OLAP cubes](#).

► To update BizTalk Databases with the new server name

1. On a computer running BizTalk Server 2010, browse to the following folder:
 - If BizTalk Server is installed on a 64-bit version of Windows Server:
%ProgramFiles(x86)%Microsoft BizTalk Server 2010\bins32\Schema\Restore
 - If BizTalk Server is installed on a 32-bit version of Windows Server:
%ProgramFiles%Microsoft BizTalk Server 2010\Schema\Restore
2. Right-click **SampleUpdateInfo.xml**, and then click **Edit**.
3. Comment out all of the database sections except for the BizTalkMgmtDb, OldPrimaryImportDatabase, PrimaryImportDatabase, ArchivingDatabase, AnalysisDatabase, StarSchemaDatabase, and Alert.
4. In the `OldPrimaryImportDatabase` section of the file, for the `ServerName` property, replace **SourceServer** with the name of existing server where the database resides.
5. In the `PrimaryImportDatabase` section of the file, for the `ServerName` property, replace **DestinationServer** with the name of the server where you have moved the BAM Primary Import database
6. For the BizTalkMgmtDb, ArchivingDatabase, AnalysisDatabase, StarSchemaDatabase, and Alert sections, set the "SourceServer" and "Destination Server" to the name of the existing server where those databases reside.



Important

Include the quotation marks around the name of the source and destination systems.



Note

If you renamed any of the BizTalk Server databases, you must also update the database names as appropriate.

7. When you are finished editing the file, save it and exit.
8. Click **Start**, click **Run**, type **cmd**, and then click **OK**.
9. At the command prompt, navigate to the following directory:
 - If BizTalk Server is installed on a 64-bit version of Windows Server:
%ProgramFiles(x86)%Microsoft BizTalk Server 2010\Schema\Restore
 - If BizTalk Server is installed on a 32-bit version of Windows Server:
%ProgramFiles%Microsoft BizTalk Server 2010\Schema\Restore
10. At the command prompt, type:
cscript UpdateDatabase.vbs SampleUpdateInfo.xml

► To update the Web.config file for the BAM portal

1. On a computer running BizTalk Server 2010, update the Web.config files under **<drive>:\Program Files\Microsoft BizTalk Server 2010\BAMPortal\BAMManagementService\Web.Config**. Update the server and database names in the following section in the Web.config:

```
<appSettings>
```

```

    <add key="BamServer" value="<ServerName>" />
    <add key="BamDatabase" value="<DatabaseName>" />
</appSettings>

```

2. On a computer running BizTalk Server 2010, update the Web.config files under **<drive>:\Program Files\Microsoft BizTalk Server 2010\BAMPortal\BAMQueryService\Web.Config**. Update the server and database names in the following section in the Web.config:

```

<appSettings>
    <add key="BamServer" value="<ServerName>" />
    <add key="BamDatabase" value="<DatabaseName>" />
    <add key="MaxResultRows" value="2000" />
</appSettings>

```

3. Save and close the files.

▶ To update reference in BAM Livedata Microsoft Excel files

1. Open the Excel live data file. The file name ends with **_LiveData.xls**.
2. On the **BAM** menu, click **BAM DB Connection**.
3. In the **Select BAM Database** dialog box, enter the SQL Server computer and the **BAMPrimaryImport** database, and then click **OK**.
4. On the **File** menu, click **Close and Return to Microsoft Excel**.
5. On the **File** menu, click **Save**.

▶ To update server and database names in all BAM SSIS packages

1. Update the server and database names in all BAM analysis SSIS packages, which are prefixed with "BAM_AN_" or "BAM_DM_". To do so, click **Start**, click **All Programs**, click **Microsoft SQL Server 2008 R2** or **Microsoft SQL Server 2008 SP1**, and then click **SQL Server Business Intelligence Development Studio**.
2. In SQL Server Business Intelligence Development Studio, create a new project. Click **File**, click **New**, and then click **Project**.
3. In the **New Project** dialog box, in the **Project Types** box, click **Business Intelligence Projects**. In the right pane, in the **Templates** box, click **Integration Services Project**, and then click **OK**.
4. In the **Integration Services Project** dialog box, in Solution Explorer, right-click **SSIS Packages**, and then click **Add Existing Package**.
5. In the **Add Copy of Existing Package** dialog box, in the **Server** drop-down list box, select the server that contains the **BAM_AN_*** and **BAM_DM_*** packages.
6. In **Package Path**, click the ellipses button.
7. In the **SSIS Package** dialog box, select the package you want to update, click **OK**, and then click **OK**.

The package is now listed in Solution Explorer.

8. In Solution Explorer, double-click the package you added in the previous step. In **Connection Managers** tab (available towards the lower half of the screen), double-click data source number 1 (BAMPrimaryImport database).
9. In the **Connection Manager** dialog box, in the **Server name** box, enter the name of the server, and then click **OK**.
10. Click the **Package Explorer** tab, double-click the **Variables** folder, and then update the values for the **PrimaryImportDatabase** and **PrimaryImportServer** variables. You must update the values to point to the new server and database.



Note

Repeat step 4 through 10 for all the packages that you want to update.

11. Click then **File** menu, and then click **Save All**.
12. Start the SQL Server Management Studio. Click **Start**, click **All Programs**, click **Microsoft SQL Server 2008 R2** or **Microsoft SQL Server 2008 SP1**, and then click **SQL Server Management Studio**.
13. In the **Connect to Server** dialog box, from the **Server** type drop-down list, select **Integration Services**.
14. Specify the server name and credentials to connect to the server and click **OK**.
15. In the **Object Explorer**, expand **Integration Services**, expand **Stored Packages**, and then click **MSDB**.
16. In the **Object Explorer Details** tab, right-click the package that you updated earlier and then click **Import Package**.
17. In the **Import Package** dialog box, from the **Package location** drop-down list, select **File System**.
18. In **Package Path**, navigate to your saved project, select the .dtsx file for the package you want to import, and then click **Open**.
19. Click inside the Package Name box to automatically populate the box.



Note

Repeat step 16 through 19 for all the packages that you want to update.

20. Click **OK**, and then click **Yes** to overwrite.
21. Enable any BAM cube update and data maintenance SSIS packages.

► **To update server and database names in data sources for all OLAP cubes**

1. Update the server and database names in data sources for all OLAP cubes. To do so, click **Start**, click **All Programs**, click **Microsoft SQL Server 2008 R2** or **Microsoft SQL Server 2008 SP1**, and then click **SQL Server Management Studio**.
2. In the **Connect to Server** dialog box, for the **Server type** drop-down list select **Analysis Services**, provide the server name, select an authentication method (and provide credentials if required), and then click **Connect**.
3. In the Object Explorer, expand **Databases**, expand **BAMAnalysis**, expand **Data**

Sources, and then double-click a data source.

4. In the **Data Source Properties** dialog box, click the ellipsis button (...) against the **Connection String** property.
5. In the **Connection Manager** dialog box, in the **Server name** box, enter the name of the server hosting the BAMPrimaryImport database, click **OK**, and then click **OK**.
6. Start all BizTalk Server services. For more information, see the topic [How To Start, Stop, Pause, Resume, or Restart BizTalk Server Services](http://go.microsoft.com/fwlink/?LinkId=154394) (<http://go.microsoft.com/fwlink/?LinkId=154394>) in BizTalk Server Help.
7. Start the IIS service.
8. Start the BAM Alerts Notification service:
 - a. Click **Start**, click **Run**, type **cmd**, and then click **OK**.
 - b. At the command prompt, type:
Net start NS\$BamAlerts
9. Resolve any incomplete trace instances. For information about resolving incomplete BAM activity instances, see [How to Resolve Incomplete Activity Instances](http://go.microsoft.com/fwlink/?LinkId=151475) (<http://go.microsoft.com/fwlink/?LinkId=151475>).

See Also

[Moving Databases](#)

How to Move the BAM Archive Database

You can use this procedure to move the BAM Archive database to another server. From an end-to-end scenario perspective, moving the BAM Archive database involves two major steps:

- [Moving the BAM Archive Database](#)
- [Updating References to the New BAM Archive Database](#)

Prerequisites

You must be logged on with an account that is a member of the SQL Server sysadmin fixed server role to perform this procedure.

Moving the BAM Archive Database

Perform the steps in the following procedure to move the BAM Archive database.

To move the BAM Archive database

1. Stop any BAM cube update and data maintenance SSIS packages, or prevent them from running until you have restored the BAM Archive database.
2. Stop all BizTalk Server services. For more information, see the topic [How To Start, Stop, Pause, Resume, or Restart BizTalk Server Services](http://go.microsoft.com/fwlink/?LinkId=154394) (<http://go.microsoft.com/fwlink/?LinkId=154394>) in BizTalk Server Help.
3. Stop the IIS service.
4. Stop the BAM Alerts Notification service:
 - a. Click **Start**, click **Run**, type **cmd**, and then click **OK**.

- b. At the command prompt, type:

Net stop NS\$BamAlerts

5. Back up the BAM Archive database on the old server. For instructions on backing up a database, follow the instructions at [How to: Back Up a Database \(SQL Server Management Studio\)](http://go.microsoft.com/fwlink/?LinkId=156510) (<http://go.microsoft.com/fwlink/?LinkId=156510>) in SQL Server Books Online on how to back up a database.
6. Copy the BAM Archive database to the new SQL Server computer.
7. Restore the BAM Archive database on the new server. For instructions on restoring the database, follow the instructions at [How to: Restore a Database Backup \(SQL Server Management Studio\)](http://go.microsoft.com/fwlink/?LinkId=156511) (<http://go.microsoft.com/fwlink/?LinkId=156511>) in SQL Server Books Online on how to restore a database.

Updating References to the New BAM Archive Database

After you have moved the database, you must update all the references to the new BAM Archive Database. The following references must be updated:

- Update the BAM configuration with the new database and server names. See [To update the BAM configuration](#).
- Update the new server and database names in all BAM analysis SSIS packages. See [To update server and database names in all BAM SSIS packages](#).

To update the BAM configuration

1. Get a copy of the .xml file used for restoring BAM:
 - a. Click **Start**, click **Run**, type **cmd**, and then click **OK**.
 - b. On a computer running BizTalk Server 2010, browse to the following folder:
 - If BizTalk Server is installed on a 64-bit version of Windows Server:
%ProgramFiles(x86)%\Microsoft BizTalk Server 2010\Tracking
 - If BizTalk Server is installed on a 32-bit version of Windows Server:
%ProgramFiles%\Microsoft BizTalk Server 2010\Tracking
 - c. At the command prompt, type:
Bm.exe get-config –filename:BAMConfiguration.xml -server:<servername> -database:<database>



Note

When running this command, substitute the actual name of the server from which to get the configuration information for <servername> and substitute the actual name of the database from which to get the configuration information for <database>. For more information about using the BAM Management (BM) utility, see [Infrastructure Management Commands](http://go.microsoft.com/fwlink/?LinkId=156516) (<http://go.microsoft.com/fwlink/?LinkId=156516>) in BizTalk Server Help.

2. Edit the BAMConfiguration.xml file and change the **ServerName** in the <DeploymentUnit

Name="ArchivingDatabase"> section to the new server name.

3. Save and close the BAMConfiguration.xml file.
4. Click **Start**, click **Run**, type **cmd**, and then click **OK**.
5. On a computer running BizTalk Server 2010, browse to the following folder:
 - If BizTalk Server is installed on a 64-bit version of Windows Server:
%ProgramFiles(x86)%\Microsoft BizTalk Server 2010\Tracking
 - If BizTalk Server is installed on a 32-bit version of Windows Server:
%ProgramFiles%\Microsoft BizTalk Server 2010\Tracking
6. At the command prompt, type:
bm.exe update-config -FileName:BAMConfiguration.xml

► To update server and database names in all BAM SSIS packages

1. Update the server and database names in all BAM analysis SSIS packages, which are prefixed with "BAM_DM_". To do so, click **Start**, click **All Programs**, click **Microsoft SQL Server 2008 R2** or **Microsoft SQL Server 2008 SP1**, and then click **SQL Server Business Intelligence Development Studio**.
2. In SQL Server Business Intelligence Development Studio, create a new project. Click **File**, click **New**, and then click **Project**.
3. In the **New Project** dialog box, in the **Project Types** box, click **Business Intelligence Projects**. On the right pane, in the **Templates** box, click **Integration Services Project**, and then click **OK**.
4. In the **Integration Services Project** dialog box, in Solution Explorer, right-click **SSIS Packages**, and then click **Add Existing Package**.
5. In the **Add Copy of Existing Package** dialog box, in the **Server** drop-down list box, select the server that contains the BAM_DM_* packages.
6. In **Package Path**, click the ellipses button.
7. In the **SSIS Package** dialog box, select the package you want to update, click **OK**, and then click **OK**.

The package is now listed in Solution Explorer.

8. In Solution Explorer, double-click the package you added in the previous step. In **Connection Managers** tab (available towards the lower half of the screen), double-click data source number 2 (BAMArchive database).
9. In the **Connection Manager** dialog box, in the **Server name** box, enter the name of the server, and then click **OK**.



Note

Repeat this for data source number 3 (MSDB database).

10. Click the **Package Explorer** tab, double-click the **Variables** folder, and then update the values for the **ArchivingDatabase**, **ArchivingServer**, **PrimaryImportDatabase**, and **PrimaryImportServer** variables. You must update the values to point to the new server and database.

**Note**

Repeat step 4 through 10 for all the packages that you want to update.

11. Click then **File** menu, and then click **Save All**.
12. Start the SQL Server Management Studio. Click **Start**, click **All Programs**, click **Microsoft SQL Server 2008 R2** or **Microsoft SQL Server 2008 SP1**, and then click **SQL Server Management Studio**.
13. In the **Connect to Server** dialog box, from the **Server** type drop-down list, select **Integration Services**.
14. Specify the server name and credentials to connect to the server and click **OK**.
15. In the **Object Explorer**, expand **Integration Services**, expand **Stored Packages**, and then click **MSDB**.
16. In the **Object Explorer Details** tab, right-click the package that you updated earlier and then click **Import Package**.
17. In the **Import Package** dialog box, from the **Package location** drop-down list, select **File System**.
18. In **Package Path**, navigate to your saved project, select the .dtsx file for the package you want to import, and then click **Open**.
19. Click inside the Package Name box to automatically populate the box.

**Note**

Repeat step 16 through 19 for all the packages that you want to update.

20. Click **OK**, and then click **Yes** to overwrite.
21. Start all BizTalk Server services. For more information, see the topic [How To Start, Stop, Pause, Resume, or Restart BizTalk Server Services](http://go.microsoft.com/fwlink/?LinkId=154394) (<http://go.microsoft.com/fwlink/?LinkId=154394>) in BizTalk Server Help.
22. Start the IIS service.
23. Start the BAM Alerts Notification service:
 - a. Click **Start**, click **Run**, type **cmd**, and then click **OK**.
 - b. At the command prompt, type:
Net start NS\$BamAlerts
24. Enable any BAM cube update and data maintenance SSIS packages.

**Tip**

As a good practice, you should also move the BAM_DM_* SSIS packages to the server hosting the BAMArchive database.

See Also

[Moving Databases](#)

How to Move the BAM Star Schema Database

You can use this procedure to move the BAM Star Schema database to another server. From an end-to-end scenario perspective, moving the BAM Star Schema database involves two major steps:

- [Moving the BAM Star Schema Database](#)
- [Updating References to the New BAM Star Schema Database](#)

Prerequisites

You must be logged on with an account that is a member of the SQL Server sysadmin fixed server role to perform this procedure.

Moving the BAM Star Schema Database

Perform the steps in the following procedure to move the BAM Star Schema database.

► To move the BAM Star Schema database

1. Stop any BAM cube update and data maintenance SSIS packages, or prevent them from running until you have restored the BAM Star Schema database.
2. Stop all BizTalk Server services. For more information, see the topic [How To Start, Stop, Pause, Resume, or Restart BizTalk Server Services](#) (<http://go.microsoft.com/fwlink/?LinkId=154394>) in BizTalk Server Help.
3. Stop the IIS service.
4. Stop the BAM Alerts Notification service:
 - a. Click **Start**, click **Run**, type **cmd**, and then click **OK**.
 - b. At the command prompt, type:
Net stop NS\$BamAlerts
5. Back up the BAM Star Schema database on the old server. For instructions on backing up a database, follow the instructions at [How to: Back Up a Database \(SQL Server Management Studio\)](#) (<http://go.microsoft.com/fwlink/?LinkId=156510>) in SQL Server Books Online on how to back up a database.
6. Copy the BAM Star Schema database to the new SQL Server computer.
7. Restore the BAM Star Schema database on the new server. For instructions on restoring the database, follow the instructions at [How to: Restore a Database Backup \(SQL Server Management Studio\)](#) (<http://go.microsoft.com/fwlink/?LinkId=156511>) in SQL Server Books Online on how to restore a database.

Updating References to the New BAM Star Schema Database

After you have moved the database, you must update all the references to the new BAM Star Schema Database. The following references must be updated:

- Update the BAM configuration with the new database and server names. See [To update the BAM configuration](#).
- Update the new server and database names in all BAM analysis SSIS packages. See [To update server and database names in all BAM SSIS packages](#).

- Update the new server and database names in data sources for all non-OLAP cubes. See [To update server and database names in data sources for all non-OLAP cubes](#).

▶ **To update the BAM configuration**

1. Get a copy of the .xml file used for restoring BAM:
 - a. Click **Start**, click **Run**, type **cmd**, and then click **OK**.
 - b. On a computer running BizTalk Server 2010, browse to the following folder:
 - If BizTalk Server is installed on a 64-bit version of Windows Server:
%ProgramFiles(x86)%\Microsoft BizTalk Server 2010\Tracking
 - If BizTalk Server is installed on a 32-bit version of Windows Server:
%ProgramFiles%\Microsoft BizTalk Server 2010\Tracking
 - c. At the command prompt, type:
Bm.exe get-config –filename:BAMConfiguration.xml -server:<servername> -database:<database>



Note

When running this command, substitute the actual name of the server from which to get the configuration information for <servername> and substitute the actual name of the database from which to get the configuration information for <database>. For more information about using the BAM Management (BM) utility, see [Infrastructure Management Commands](#) (<http://go.microsoft.com/fwlink/?LinkId=156516>) in BizTalk Server Help.

2. Edit the BAMConfiguration.xml file and change the **ServerName** in the <DeploymentUnit Name="StarSchemaDatabase"> section to the new server name.
3. Save and close the BAMConfiguration.xml file.
4. Click **Start**, click **Run**, type **cmd**, and then click **OK**.
5. On a computer running BizTalk Server 2010, browse to the following folder:
 - If BizTalk Server is installed on a 64-bit version of Windows Server:
%ProgramFiles(x86)%\Microsoft BizTalk Server 2010\Tracking
 - If BizTalk Server is installed on a 32-bit version of Windows Server:
%ProgramFiles%\Microsoft BizTalk Server 2010\Tracking
6. At the command prompt, type:
bm.exe update-config -FileName:BAMConfiguration.xml

▶ **To update server and database names in all BAM SSIS packages**

1. Update the server and database names in all BAM analysis SSIS packages, which are prefixed with "BAM_AN_". To do so, click **Start**, click **All Programs**, click **Microsoft SQL Server 2008 R2** or **Microsoft SQL Server 2008 SP1**, and then click **SQL Server Business Intelligence Development Studio**.

2. In SQL Server Business Intelligence Development Studio, create a new project. Click **File**, click **New**, and then click **Project**.
3. In the **New Project** dialog box, in the **Project Types** box, click **Business Intelligence Projects**. On the right pane, in the **Templates** box, click **Integration Services Project**, and then click **OK**.
4. In the **Integration Services Project** dialog box, in Solution Explorer, right-click **SSIS Packages**, and then click **Add Existing Package**.
5. In the **Add Copy of Existing Package** dialog box, in the **Server** drop-down list box, select the server that contains the BAM_AN_* packages.
6. In **Package Path**, click the ellipses button.
7. In the **SSIS Package** dialog box, select the package you want to update, click **OK**, and then click **OK**.

The package is now listed in Solution Explorer.

8. In Solution Explorer, double-click the package you added in the previous step. In **Connection Managers** tab (available towards the lower half of the screen), double-click data source number 2 (BAMStarSchema database).
9. In the **Connection Manager** dialog box, in the **Server name** box, enter the name of the server, and then click **OK**.

 **Note**

Repeat this for data source number 3 (MSDB database).

10. In the **Connection Managers** tab, double-click data source number 4 (BAMAnalysis database). In the **Add Analysis Services Connection Manager** dialog box, click **Edit**.
11. In the **Connection Manager** dialog box, in the **Server name** box, enter the name of the server, click **OK**, and then click **OK**.
12. Click the **Package Explorer** tab, double-click the **Variables** folder, and then update the values for the **AnalysisDatabase**, **AnalysisServer**, **PrimaryImportDatabase**, **PrimaryImportServer**, **StarSchemaDatabase**, and **StarSchemaServer** variables. You must update the values to point to the new server and database.

 **Note**

Repeat step 4 through 12 for all the packages that you want to update.

13. Click then **File** menu, and then click **Save All**.
14. Start the SQL Server Management Studio. Click **Start**, click **All Programs**, click **Microsoft SQL Server 2008 R2** or **Microsoft SQL Server 2008 SP1**, and then click **SQL Server Management Studio**.
15. In the **Connect to Server** dialog box, from the **Server type** drop-down list, select **Integration Services**.
16. Specify the server name and credentials to connect to the server and click **OK**.
17. In the **Object Explorer**, expand **Integration Services**, expand **Stored Packages**, and then click **MSDB**.
18. In the **Object Explorer Details** tab, right-click the package that you updated earlier and then click **Import Package**.

19. In the **Import Package** dialog box, from the **Package location** drop-down list, select **File System**.
20. In **Package Path**, navigate to your saved project, select the .dtsx file for the package you want to import, and then click **Open**.
21. Click inside the Package Name box to automatically populate the box.



Repeat step 18 through 21 for all the packages that you want to update.

22. Click **OK**, and then click **Yes** to overwrite.
23. Enable any BAM cube update and data maintenance SSIS packages.

► **To update server and database names in data sources for all non-OLAP cubes**

1. Update the server and database names in data sources for all non-OLAP cubes. To do so, click **Start**, click **All Programs**, click **Microsoft SQL Server 2008 R2** or **Microsoft SQL Server 2008 SP1**, and then click **SQL Server Management Studio**.
2. In the **Connect to Server** dialog box, for the **Server type** drop-down list select **Analysis Services**, provide the server name, select an authentication method (and provide credentials if required), and then click **Connect**.
3. In the Object Explorer, expand **Databases**, expand **BAMAnalysis**, expand **Data Sources**, and then double-click a data source.
4. In the **Data Source Properties** dialog box, click the ellipsis button (...) against the **Connection String** property.
5. In the **Connection Manager** dialog box, in the **Server name** box, enter the name of the server hosting the BAMStarSchema database, click **OK**, and then click **OK**.
6. Start all BizTalk Server services. For more information, see the topic [How To Start, Stop, Pause, Resume, or Restart BizTalk Server Services](http://go.microsoft.com/fwlink/?LinkId=154394) (<http://go.microsoft.com/fwlink/?LinkId=154394>) in BizTalk Server Help.
7. Start the IIS service.
8. Start the BAM Alerts Notification service:
 - a. Click **Start**, click **Run**, type **cmd**, and then click **OK**.
 - b. At the command prompt, type:
Net start NS\$BamAlerts
9. Resolve any incomplete trace instances. For information about resolving incomplete BAM activity instances, see [How to Resolve Incomplete Activity Instances](http://go.microsoft.com/fwlink/?LinkId=151475) (<http://go.microsoft.com/fwlink/?LinkId=151475>).



As a good practice, you should also move the BAM_AN_* SSIS packages to the server hosting the BAMStarSchema database.

See Also

[Moving Databases](#)

How to Move the BAM Analysis Database

You can use this procedure to move the BAM Analysis database to another server. From an end-to-end scenario perspective, moving the BAM Analysis database involves two major steps:

- [Moving the BAM Analysis Database](#)
- [Updating References to the New BAM Analysis Database](#)

Prerequisites

You must be logged on with an account that is a member of the SQL Server sysadmin fixed server role to perform this procedure.

Moving the BAM Analysis Database

Perform the steps in the following procedure to move the BAM Analysis database.

▶ To move the BAM Analysis database

1. Stop any BAM cube update and data maintenance SSIS packages, or prevent them from running until you have restored the BAM Analysis database.
2. Stop all BizTalk Server services. For more information, see the topic [How To Start, Stop, Pause, Resume, or Restart BizTalk Server Services](#) (<http://go.microsoft.com/fwlink/?LinkId=154394>) in BizTalk Server Help.
3. Stop the IIS service.
4. Stop the BAM Alerts Notification service:
 - a. Click **Start**, click **Run**, type **cmd**, and then click **OK**.
 - b. At the command prompt, type:
Net stop NS\$BamAlerts
5. Back up the BAM Analysis database on the old server. For instructions on backing up a database, follow the instructions at [How to: Back Up a Database \(SQL Server Management Studio\)](#) (<http://go.microsoft.com/fwlink/?LinkId=156510>) in SQL Server Books Online on how to back up a database.
6. Copy the BAM Analysis database to the new SQL Server computer.
7. Restore the BAM Analysis database on the new server. For instructions on restoring the database, follow the instructions at [How to: Restore a Database Backup \(SQL Server Management Studio\)](#) (<http://go.microsoft.com/fwlink/?LinkId=156511>) in SQL Server Books Online on how to restore a database.

Updating References to the New BAM Analysis Database

After you have moved the database, you must update all the references to the new BAM Analysis Database. The following references must be updated:

- Update the BAM configuration with the new database and server names. See [To update the BAM configuration](#).
- Update the new server and database names in all BAM analysis SSIS packages. See [To update server and database names in all BAM SSIS packages](#).

▶ To update the BAM configuration

1. Get a copy of the .xml file used for restoring BAM:
 - a. Click **Start**, click **Run**, type **cmd**, and then click **OK**.
 - b. On a computer running BizTalk Server 2010, browse to the following folder:
 - If BizTalk Server is installed on a 64-bit version of Windows Server:
%ProgramFiles(x86)%\Microsoft BizTalk Server 2010\Tracking
 - If BizTalk Server is installed on a 32-bit version of Windows Server:
%ProgramFiles%\Microsoft BizTalk Server 2010\Tracking
 - c. At the command prompt, type:
Bm.exe get-config -filename:BAMConfiguration.xml -server:<servername> - database:<database>



Note

When running this command, substitute the actual name of the server from which to get the configuration information for <servername> and substitute the actual name of the database from which to get the configuration information for <database>. For more information about using the BAM Management (BM) utility, see [Infrastructure Management Commands](http://go.microsoft.com/fwlink/?LinkId=156516) (<http://go.microsoft.com/fwlink/?LinkId=156516>) in BizTalk Server Help.

2. Edit the BAMConfiguration.xml file and change the **ServerName** in the `<DeploymentUnit Name="AnalysisDatabase">` section to the new server name.
3. Save and close the BAMConfiguration.xml file.
4. Click **Start**, click **Run**, type **cmd**, and then click **OK**.
5. On a computer running BizTalk Server 2010, browse to the following folder:
 - If BizTalk Server is installed on a 64-bit version of Windows Server:
%ProgramFiles(x86)%\Microsoft BizTalk Server 2010\Tracking
 - If BizTalk Server is installed on a 32-bit version of Windows Server:
%ProgramFiles%\Microsoft BizTalk Server 2010\Tracking
6. At the command prompt, type:
bm.exe update-config -FileName:BAMConfiguration.xml

► To update server and database names in all BAM SSIS packages

1. Update the server and database names in all BAM analysis SSIS packages, which are prefixed with "BAM_AN_". To do so, click **Start**, click **All Programs**, click **Microsoft SQL Server 2008 R2** or **Microsoft SQL Server 2008 SP1**, and then click **SQL Server Business Intelligence Development Studio**.
2. In SQL Server Business Intelligence Development Studio, create a new project. Click **File**, click **New**, and then click **Project**.
3. In the **New Project** dialog box, in the **Project Types** box, click **Business Intelligence Projects**. On the right pane, in the **Templates** box, click **Integration Services Project**,

and then click **OK**.

4. In the **Integration Services Project** dialog box, in Solution Explorer, right-click **SSIS Packages**, and then click **Add Existing Package**.
5. In the **Add Copy of Existing Package** dialog box, in the **Server** drop-down list box, select the server that contains the BAM_AN_* packages.
6. In **Package Path**, click the ellipses button.
7. In the **SSIS Package** dialog box, select the package you want to update, click **OK**, and then click **OK**.

The package is now listed in Solution Explorer.

8. In Solution Explorer, double-click the package you added in the previous step. In **Connection Managers** tab (available towards the lower half of the screen), double-click data source number 2 (BAMArchive database).
9. In the **Connection Manager** dialog box, in the **Server name** box, enter the name of the server, and then click **OK**.

 **Note**

Repeat this for data source number 3 (MSDB database).

10. In the **Connection Managers** tab, double-click data source number 4 (BAMAnalysis database). In the **Add Analysis Services Connection Manager** dialog box, click **Edit**.
11. In the **Connection Manager** dialog box, in the **Server name** box, enter the name of the server, click **OK**, and then click **OK**.
12. Click the **Package Explorer** tab, double-click the **Variables** folder, and then update the values for the **AnalysisDatabase**, **AnalysisServer**, **PrimaryImportDatabase**, **PrimaryImportServer**, **StarSchemaDatabase**, and **StarSchemaServer** variables. You must update the values to point to the new server and database.

 **Note**

Repeat step 4 through 12 for all the packages that you want to update.

13. Click then **File** menu, and then click **Save All**.
14. Start the SQL Server Management Studio. Click **Start**, click **All Programs**, click **Microsoft SQL Server 2008 R2** or **Microsoft SQL Server 2008 SP1**, and then click **SQL Server Management Studio**.
15. In the **Connect to Server** dialog box, from the **Server** type drop-down list, select **Integration Services**.
16. Specify the server name and credentials to connect to the server and click **OK**.
17. In the **Object Explorer**, expand **Integration Services**, expand **Stored Packages**, and then click **MSDB**.
18. In the **Object Explorer Details** tab, right-click the package that you updated earlier and then click **Import Package**.
19. In the **Import Package** dialog box, from the **Package location** drop-down list, select **File System**.
20. In **Package Path**, navigate to your saved project, select the .dtsx file for the package you

want to import, and then click **Open**.

21. Click inside the Package Name box to automatically populate the box.

 **Note**

Repeat step 18 through 21 for all the packages that you want to update.

22. Click **OK**, and then click **Yes** to overwrite.

23. Start all BizTalk Server services. For more information, see the topic [How To Start, Stop, Pause, Resume, or Restart BizTalk Server Services](#) (<http://go.microsoft.com/fwlink/?LinkId=154394>) in BizTalk Server Help.

24. Start the IIS service.

25. Start the BAM Alerts Notification service:

- a. Click **Start**, click **Run**, type **cmd**, and then click **OK**.
- b. At the command prompt, type:

Net start NS\$BamAlerts

26. Enable any BAM cube update and data maintenance SSIS packages.

See Also

[Moving Databases](#)

How to Move the BAM Notification Services Databases

You can use this procedure to move the BAM Notification Services database to another server. From an end-to-end scenario perspective, moving the BAM Notification Services database involves two major steps:

- [Moving the BAM Notification Services Database](#)
- [Updating References to the New BAM Notification Services Databases](#)

 **Note**

You must move the BAM Notification Services Application (BAMAlertsApplication) database and the BAM Notification Services Instance (BAMAlertsNSMain) database together.

Prerequisites

You must be logged on with an account that is a member of the SQL Server sysadmin fixed server role to perform this procedure.

Moving the BAM Notification Services Database

Perform the steps in the following procedure to move the BAM Notification Services database.

To move the BAM Notification Services database

1. Stop any BAM cube update and data maintenance SSIS packages, or prevent them from running until you have restored the BAM Notification Services database.
2. Stop all BizTalk Server services. For more information, see the topic [How To Start, Stop, Pause, Resume, or Restart BizTalk Server Services](#)

(<http://go.microsoft.com/fwlink/?LinkId=154394>) in BizTalk Server Help.

3. Stop the IIS service.
4. Stop the BAM Alerts Notification service:
 - a. Click **Start**, click **Run**, type **cmd**, and then click **OK**.
 - b. At the command prompt, type:
Net stop NS\$BamAlerts
5. Back up the BAM Notification Services database on the old server. For instructions on backing up a database, follow the instructions at [How to: Back Up a Database \(SQL Server Management Studio\)](#) (<http://go.microsoft.com/fwlink/?LinkId=156510>) in SQL Server Books Online on how to back up a database.



Note

Perform this step for both BAMAlertsApplication and BAMAlertsNSMain databases.

6. Copy the BAM Notification Services database to the new SQL Server computer.
7. Restore the BAM Notification Services database on the new server. For instructions on restoring the database, follow the instructions at [How to: Restore a Database Backup \(SQL Server Management Studio\)](#) (<http://go.microsoft.com/fwlink/?LinkId=156511>) in SQL Server Books Online on how to restore a database.



Note

Perform this step for both BAMAlertsApplication and BAMAlertsNSMain databases.

Updating References to the New BAM Notification Services Databases

After you have moved the database, you must update all the references to the new BAM Notification Services databases. The following references must be updated:

- Update the BAM configuration with the new database and server names. See [To update the BAM configuration](#).
- Re-register the Notification Service on all computers in the BizTalk Server group. See [Register the Notification Services](#).

▶ To update the BAM configuration

1. Get a copy of the .xml file used for restoring BAM:
 - a. Click **Start**, click **Run**, type **cmd**, and then click **OK**.
 - b. On a computer running BizTalk Server 2010, browse to the following folder:
 - If BizTalk Server is installed on a 64-bit version of Windows Server:
%ProgramFiles(x86)%\Microsoft BizTalk Server 2010\Tracking
 - If BizTalk Server is installed on a 32-bit version of Windows Server:
%ProgramFiles%\Microsoft BizTalk Server 2010\Tracking

- c. At the command prompt, type:

Bm.exe get-config –filename:BAMConfiguration.xml -server:<servername> - database:<database>



Note

When running this command, substitute the actual name of the server from which to get the configuration information for <servername> and substitute the actual name of the database from which to get the configuration information for <database>. For more information about using the BAM Management (BM) utility, see [Infrastructure Management Commands](#) (<http://go.microsoft.com/fwlink/?LinkId=156516>) in BizTalk Server Help.

2. Edit the BAMConfiguration.xml file and change the **DBServer** properties in the <DeploymentUnit Name="Alert"> section to the new server name.
3. Save and close the BAMConfiguration.xml file.
4. Click **Start**, click **Run**, type **cmd**, and then click **OK**.
5. On a computer running BizTalk Server 2010, browse to the following folder:
 - If BizTalk Server is installed on a 64-bit version of Windows Server:
%ProgramFiles(x86)%Microsoft BizTalk Server 2010\Tracking
 - If BizTalk Server is installed on a 32-bit version of Windows Server:
%ProgramFiles%Microsoft BizTalk Server 2010\Tracking
6. At the command prompt, type:
bm.exe update-config -FileName:BAMConfiguration.xml

Register the Notification Services

After you have moved the BAM Notification Services database, you must re-register the Notification Service on all computers in the BizTalk Server group that are running Notification Services (NSservice.exe). This enables Notification Services to connect to the databases in their new location. For instructions on how to register the Notification Services, follow steps 5 through 11 at [How to Update References to the BAM Notification Services Databases](#) (<http://go.microsoft.com/fwlink/?LinkId=156684>) in BizTalk Server Help.

Note the following while performing steps mentioned in the preceding link:

- Steps 5 and 6 in the preceding link must be performed on the servers listed in the BAM configuration XML for the following properties:

```
<DeploymentUnit Name="Alert">
  <Property Name="GeneratorServerName">Server_Name</Property>
  <Property Name="ProviderServerName">Server_Name</Property>
  <Property Name="DistributorServerName">Server_Name</Property>
</DeploymentUnit>
```

- Step 7 through 11 must be performed on the computer that hosts the BAM portal.

See Also

[Moving Databases](#)

Moving Non-BAM Databases

You can use this procedure to move the primary BizTalk Server databases to another server. This same basic procedure can also be used to move the BizTalk Server databases from a local SQL Server to a remote SQL Server or to a SQL Server cluster. This section describes how to move the BizTalk Server databases that are not BAM related.

In This Section

To move the non BAM databases follow the steps in the topic [How to Move the BizTalk Server Databases](#) (<http://go.microsoft.com/fwlink/?LinkId=210646>) in BizTalk Server help.

This section also contains a topic that describes procedures that must be followed after moving particular BizTalk Server databases. Complete the steps in the topic as appropriate for your environment.

- [Considerations When Moving the Tracking Database if the MessageBox Database Is Not Being Moved](#)

Considerations When Moving the Tracking Database if the MessageBox Database Is Not Being Moved

If you are moving the Tracking database but not the MessageBox database, when you edit the SampleUpdateInfo.xml file, make sure that you include an entry for the MessageBox database as well, even though the MessageBox database is not being moved. This must be done to ensure that the SQL Server Agent job TrackedMessages_Copy_BizTalkMsgBoxDb is updated with the location of the new Tracking database.

For example, in the following SampleUpdateInfo.xml file, only the Tracking database is being moved from OldServer to NewServer. The MessageBox database is staying on OldServer:

```
<UpdateConfiguration>
    <MessageBoxDB oldDBName="BizTalkMgmtDb" oldDBServer="OldServer"
newDBName="BizTalkMgmtDb" newDBServer="OldServer" IsMaster="1"/>
    <TrackingDB oldDBName="BizTalkDTADB" oldDBServer="OldServer"
newDBName="BizTalkDTADB" newDBServer="NewServer"/>
    <OtherDatabases>
    </OtherDatabases>
</UpdateConfiguration>
```

See Also

[Moving Databases](#)

Monitoring BizTalk Server

This section describes essential tasks for monitoring BizTalk applications and infrastructure. Monitoring BizTalk Server on a regular basis and resolving any issues that you find helps to keep your BizTalk applications accessible. The goal of monitoring is to minimize the amount of time that an issue goes undetected and therefore, unresolved. Additionally, you can use monitoring to help detect situations that might cause an issue.

In This Section

- [Monitoring the BizTalk Server Environment](#)
- [Best Practices for Monitoring](#)
- [Routine Monitoring Tasks](#)
- [Using the Performance Analysis of Logs \(PAL\) Tool](#)
- [Monitoring BizTalk Server with System Center Operations Manager 2007](#)

Monitoring the BizTalk Server Environment

You can monitor BizTalk Server infrastructure and applications with a manual or automatic process, or a combination of the two methods, using the tools as shown in the following table.

Manual or Automated Monitoring	Tools
Automated Monitoring	<ul style="list-style-type: none">• Microsoft System Center Operations Manager (Operations Manager)
Manual Monitoring	<ul style="list-style-type: none">• The Group Hub page in the BizTalk Server Administration console• Performance Analysis of Logs (PAL) tool• Event Viewer

Whether or not you implement a monitoring application, you should use the BizTalk Server Administration console to monitor the health of your BizTalk Server applications and perform root-cause analysis to identify the underlying cause of any problems.

When monitoring BizTalk Server, keep these points in mind:

- Your infrastructure could be healthy, but your applications might not be (for example, they are receiving invalid messages and are unable to process them).
- Your infrastructure could be unhealthy, but your applications might be running fine (for example, if a server is down, but there are enough servers assigned to the host to take over the load).
- An infrastructure problem could surface as an application problem (for example, messages are not being processed fast enough because a server is down).

Monitoring Types

Monitoring your BizTalk Server and applications falls into four main categories:

- Availability monitoring
- Health monitoring
- Performance monitoring
- Threshold monitoring

Availability Monitoring

Availability monitoring answers the question "Is the unavailability of a system or application resource preventing your BizTalk Server applications from running optimally?" These issues are almost exclusively system-level, such as availability of services and connections. For example, if an adapter is failing because the Enterprise Single Sign-On service is stopped, this is an availability issue. If one of the servers assigned to a host has failed and your application is falling behind on processing messages, you have an availability issue. Likewise, if an application is stopped and is unable to process messages, you have an availability issue. The following table lists the availability monitoring tools.

Tool	Task
BizTalk Server Administration console	Check the Group Hub page in the BizTalk Server Administration console to see if applications or their components (ports/orchestrations) are stopped.
Operations Manager 2007	The BizTalk Server management pack and Operations Manager Operations console displays alerts if critical low-level services such as adapters are unavailable. To effectively monitor BizTalk Server, you must monitor non-BizTalk Server resources that your applications depend on, such as databases and servers. In addition, you must also install and use the SQL Server, Internet Information Services, and Windows Base Operating System Management Packs. Operations Manager consolidates events of interest from event logs, WMI, and other event providers. For more information about installing all the relevant management packs, see Checklist: Monitoring BizTalk Server with Operations Manager 2007 .
Event Viewer	Look for adapter connection issues, stopped services, and so on.

Health Monitoring

Health monitoring helps you answer the question, "Are any of my applications or resources in bad health?" For example, are any of my applications or their constituent artifacts currently experiencing exception conditions? Or, are messages suspended because of invalid data in the message payload? The following table shows health-monitoring tools.

Tool	Task
BizTalk Server Administration console	You use the Group Hub page and query pages in the BizTalk Server Administration console to identify application health problems and analyze their cause(s).
Operations Manager	The BizTalk Server management pack is your first line of defense to notify you that you have suspended messages and/or service instances in your BizTalk Server applications. After receiving notification from Operations Manager, you can transition to the BizTalk Server Administration console to troubleshoot the problem.
Event Viewer	Detect problems that occur during the processing of messages and orchestrations.

Performance Monitoring

Performance monitoring answers the question, "How efficiently is the system performing its work?" This kind of monitoring focuses primarily on the load on physical resources like databases and disks. For example, if the CPU utilization is consistently at 90 to 100 percent and a backlog of messages is forming, this is a performance issue at the computer level. The following table shows performance-monitoring tools.

Tool	Task
SQL Query Analyzer	Monitor database size and content to diagnose system problems.
Operations Manager	The BizTalk Server management pack and Operations Manager Operations console can be configured to display alerts if critical BizTalk Server performance counters such as the Message Box Q size or Host Q size exceed the defined thresholds. To monitor the performance of non-BizTalk Server resources that your

Tool	Task
	<p>applications depend on, such as databases and servers, you must also install and use the SQL Server, Internet Information Services, and Windows Base OS Management Packs. For more information about installing all the relevant management packs, see Checklist: Monitoring BizTalk Server with Operations Manager 2007.</p> <p>You can also use the Performance Analysis of Logs (PAL) tool to capture the threshold values from the throughput testing to use in the threshold rules in the BizTalk Server management pack. For more information about the PAL tool, see Using the Performance Analysis of Logs (PAL) Tool.</p>
BizTalk Server Administration console	The Group Hub page shows key performance metrics such as the number of service instances currently active, dehydrated, ready to run, scheduled, suspended, etc. in your BizTalk Server applications.
Business Activity Monitoring (BAM)	You can specify specific stages in your business process for which you want to track key performance indicators pertinent to your business application. Using BAM, you can monitor business metrics as well as IT metrics (for example, SLA's and execution times).

Threshold Monitoring

Customized threshold rules are an essential element in a mature operations environment. You can create many of these threshold rules in Operations Manager. These threshold rules are typically based on the requirements of the BizTalk application. The Performance Analysis of Logs (PAL) tool can streamline the process of determining the correct values for these thresholds for your environment. The PAL tool comes with some base threshold values that can serve as the core of the data that is used for Microsoft System Center Operations Manager. Implementing those threshold rules in Operations Manager allows for automated monitoring. In addition, an administrator can setup notification rules and can perform actions based on the triggering of a threshold rule (such as running a script, calling .NET code, sending e-mail, etc.). The following table shows threshold-monitoring tools.

Tool	Task
Performance Analysis of Logs (PAL) tool	The PAL tool automatically reports when performance counters are beyond thresholds. The thresholds dynamically change to be appropriate for the environment of the server. For example, the kernel memory pool thresholds change based on the answers that the user provides about 32-bit/64-bit architecture, amount of physical memory, and /3GB switch. The PAL tool can be downloaded for free at http://www.codeplex.com/PAL .
Operations Manager	The BizTalk Server management pack and Operation Manager Operations console can be configured to display alerts if critical BizTalk Server counters exceed the defined thresholds.

Troubleshooting

Once you are aware of a health problem with your BizTalk Server applications, you can use the **Group Hub** page and **Query** pages in the BizTalk Server Administration console to analyze the problem. The BizTalk Server Administration console provides an integrated configuration, deployment and troubleshooting experience, and you can fix configuration and deployment related problems within the Administration console after you have pinpointed them. Typically, most application problems are due to messages not getting through as expected (this can manifest as suspended service instances, or retrying ports, or dehydrated instances that have not been reactivated, etc.)

You can use the **Group Hub** page and **Query** pages to group your service instances (whatever state they are in: running, suspended, dehydrated, etc.) by application, error type, service type, host, etc., to isolate the different errors, investigate them one by one, and fix them.

Best Practices for Monitoring

This topic provides best practices for monitoring your Microsoft BizTalk Server environment and applications.

Create and then implement a monitoring plan for your BizTalk applications and infrastructure

- Read the monitoring topics in this guide to ensure a more complete monitoring solution. Factors to consider include the following:
 - Who will perform the daily, weekly, monthly, and as needed monitoring tasks?
 - Is someone notified of events, suspended messages, or other system or application failures?

- Are "expected" exceptions filtered or given a lower priority?
- Are all host instances monitored to ensure they continue running?
- Are all custom services, custom event logs, and custom databases monitored?
- Are the SQL Server computers and the BizTalk Server SQL agent jobs monitored?

If possible, install a monitoring application such as Operations Manager 2007 in order to automate the monitoring of your BizTalk Server applications and infrastructure

- Using Microsoft System Center Operations Manager is the preferred approach for automated monitoring because the BizTalk Server management packs provide hundreds of built-in rules for BizTalk Server.

For more information, see the following resources:

- [Microsoft BizTalk Server Management Pack for System Center Operations Manager 2007](http://go.microsoft.com/fwlink/?LinkID=190339) (<http://go.microsoft.com/fwlink/?LinkID=190339>).
- [How to Import a Management Pack in Operations Manager 2007](http://go.microsoft.com/fwlink/?LinkID=98348) (<http://go.microsoft.com/fwlink/?LinkID=98348>)
- [How to Mark BizTalk Server Databases for Customized Monitoring](#)

Run the BizTalk Server Best Practices Analyzer

- The BizTalk Server Best Practices Analyzer examines a BizTalk Server deployment and generates a list of issues pertaining to best practices standards. The tool performs configuration-level verification by gathering data from different information sources, such as Windows Management Instrumentation (WMI) classes, SQL Server databases, and registry entries. The data is then used to evaluate the deployment configuration. The tool reads and reports only and does not modify any system settings, and is not a self-tuning tool.

You can download the BizTalk Server Best Practices Analyzer at <http://go.microsoft.com/fwlink/?LinkId=83317> (<http://go.microsoft.com/fwlink/?LinkId=83317>).

Run the Performance Analysis of Logs tool (PAL)

- PAL is available as a free download at <http://go.microsoft.com/fwlink/LinkID=98098>. For important installation information, see [Using the Performance Analysis of Logs \(PAL\) Tool](#).

Run Log Parser

- Log Parser is a powerful, versatile tool that provides universal query access to text-based data such as log files, XML files and CSV files, as well as key data sources on the Windows® operating system such as the Event Log, the Registry, the file system, and Active Directory®. You may want to use this tool to query a significant amount of logging information. You can download the Log Parser tool at <http://go.microsoft.com/fwlink/?LinkID=85574>

Run the BizTalk MsgBoxViewer Tool

Run the [BizTalk MsgBoxViewer tool](#) available from <http://go.microsoft.com/fwlink/?LinkId=151930>. This tool analyzes the BizTalk MessageBox and other databases and generates an HTML report containing warnings, if any, and other information related to the databases. You can also save the reports for later use. These reports might be useful when troubleshooting issues with the BizTalk application.

If the BizTalk MsgBoxViewer tool identifies any issues, run the [Terminator tool](#) available at <http://go.microsoft.com/fwlink/?LinkId=151931>. This tool enables users to easily resolve any

issues identified by the BizTalk MsgBoxViewer tool. For more information about how the Terminator tool integrates with the BizTalk MsgBoxViewer tool, see [Using BizTalk Terminator to resolve issues identified by BizTalk MsgBoxViewer](http://go.microsoft.com/fwlink/?LinkId=151932) (<http://go.microsoft.com/fwlink/?LinkId=151932>).



Note

Use of these tools is not supported by Microsoft, and Microsoft makes no guarantees about the suitability of these programs. Use of these programs is entirely at your own risk.

Make monitoring a priority

- Consistent monitoring of BizTalk Server applications and infrastructure is essential to maintaining a healthy environment.
- Regularly evaluate and adjust your monitoring tools over time and as your BizTalk Server applications and infrastructure change.

Routine Monitoring Tasks

Performing the following monitoring tasks on a regularly scheduled basis will assist you in keeping your BizTalk Server applications and infrastructure operationally ready.

Daily Monitoring Tasks

- Review all open alerts.
- Use the **Group Hub** page in the BizTalk Server Administration Console to investigate orchestration, port, and message failures. The **Group Hub** page provides access to the current real-time state of the system, accessing data in the MessageBox database. You can view all service instances such as orchestrations, ports, and messaging, along with their associated messages. Using the **Group Hub** page you can perform the following activities:
 - See currently running service instances, such as orchestrations and messaging, and their associated messages.
 - Look into the MessageBox database for a view of the current data and the real-time state of the system.
 - Suspend, terminate, and resume service instances.

For more information about using the **Group Hub** page, see <http://go.microsoft.com/fwlink/?LinkId=155281>.

- Review warnings (optional).

For more information, see [Checklist: Performing Daily Maintenance Checks](#).

Weekly Monitoring Tasks

- Review the event logs at least once per week. The reason for this task is to prevent issues, such as DBNetLib errors going undetected. Service interruption might go unnoticed unless you have a very low latency system. However, some of these errors can indicate a bigger

issue (for example, too many hosts or BizTalk Server servers for the number of message boxes, performance issues on the SQL box, etc).

For more information, see [Checklist: Performing Weekly Maintenance Checks](#).

As-Needed Tasks

- Modify the rules to customize the monitoring of your BizTalk Server applications and infrastructure.
- Run the Performance Analysis of Logs tool (PAL). If your BizTalk Server deployment is fairly constant (for example, new trading partners aren't added routinely, new code is not deployed), you might run Perfmon and PAL once a quarter or even every six months. If your BizTalk Server deployment changes more frequently, you may want to run Perfmon and PAL every couple of months to compare against a baseline. For more information on PAL, see [Using the Performance Analysis of Logs \(PAL\) Tool](#).
- Run Perfmon every few months, to once a quarter or even every six months depending on the number of changes occur in your BizTalk Server deployment.
- Run BizTalk Server Best Practices Analyzer when the BizTalk Server deployment changes (for example, the addition of new applications or creation of new hosts). You can download the BizTalk Server Best Practices Analyzer at <http://go.microsoft.com/fwlink/?LinkId=83317>.
- Run the [BizTalk MsgBoxViewer tool](#) (<http://go.microsoft.com/fwlink/?LinkId=151930>). This tool analyzes the BizTalk MessageBox and other databases and generates an HTML report containing warnings, if any, and other information related to the databases.

Tip

You can also save the reports for later use. These reports might be useful when troubleshooting issues with the BizTalk application.

Note

Use of this tool is not supported by Microsoft, and Microsoft makes no guarantees about the suitability of this programs. Use of this program is entirely at your own risk.

- Run the [Terminator tool](#) (<http://go.microsoft.com/fwlink/?LinkId=151931>). This tool enables users to easily resolve any issues identified by the BizTalk MsgBoxViewer tool. For more information about how the Terminator tool integrates with the BizTalk MsgBoxViewer tool, see [Using BizTalk Terminator to resolve issues identified by BizTalk MsgBoxViewer](#) (<http://go.microsoft.com/fwlink/?LinkId=151932>).

Note

Use of this tool is not supported by Microsoft, and Microsoft makes no guarantees about the suitability of this programs. Use of this program is entirely at your own risk.

Annual Monitoring Tasks

- Review the effectiveness of the monitoring of your BizTalk Server applications and infrastructure.
- Create a plan to make any needed changes in monitoring.

See Also

[Routine Maintenance Checklists](#)

Using the Performance Analysis of Logs (PAL) Tool

The PAL (Performance Analysis of Logs) tool reads in a performance monitor counter log (any known format) and analyzes it using complex, but known thresholds (provided). The tool generates an HTML based report that graphically charts important performance counters and throws alerts when thresholds are exceeded. The thresholds are originally based on thresholds defined by the Microsoft product teams, including BizTalk Server, and members of Microsoft support. This tool is not a replacement of traditional performance analysis, but it automates the analysis of performance counter logs enough to help save you time. The PAL tool:

- Analyzes performance counter logs for thresholds
- Is helpful for large Perfmon logs
- Identifies BizTalk Server and operating system performance counter bottlenecks by analyzing for thresholds
- Is extensible to do analysis on any performance counters
- Can be used to help write your own counter

PAL is available as a free download at <http://go.microsoft.com/fwlink/?linkid=98098>. It requires Microsoft Log Parser. Log Parser is a powerful, versatile tool that provides universal query access to text-based data such as log files, XML files, and CSV files, as well as key data sources on the Windows operating system such as the event log, the registry, the file system, and Active Directory® directory service. You may want to use this tool to query a significant amount of logging information. You can download the Log Parser tool at <http://go.microsoft.com/fwlink/?linkid=85574>.

Using PAL with Performance Counter Logs in Different Languages

The PAL tool analyzes performance counter logs only in English language. To use the PAL tool with performance counter logs in other languages, you must first translate the logs to English language. You can use the **Perfmon Log Translator** tool available at <http://go.microsoft.com/fwlink/?LinkId=158802> to translate the original performance counter log files to English.

Understanding the PAL Tool Report for Microsoft BizTalk Server 2010

The PAL tool provides Perfmon log analysis of thresholds for the Windows operating system, Microsoft SQL Server, and BizTalk Server. This section describes most of the analyses in the BizTalk Server 2010 threshold report in the PAL tool.



Note

This topic is long so that comprehensive information about the PAL tool can be contained in one place for easy reference.

The following analysis and thresholds are reported by the PAL tool.

Analysis Type and Name	Analysis Description
Disk: Disk Free Space for a Kernel Dump	This analysis checks to make sure there is enough free disk space for the operating system to dump all memory to disk. If insufficient disk space is available, then the operating system will fail to create a memory.dmp, file which is necessary to analyze the root cause of a kernel dump.
Disk: Logical/Physical Disk Interface Analysis	This analysis looks at the idle time of each of the physical disks. The more idle the disk is, the less the disk is being used. This counter is best used when one disk is used in the logical disk. “% Idle Time” reports the percentage of time during the sample interval that the disk was idle. Reference: Ruling Out Disk-Bound Problems at http://go.microsoft.com/fwlink/?linkid=50669
Disk: Logical/Physical Disk Read/Write Latency Analysis	The most reliable way for Windows to detect a disk performance bottleneck is by measuring its response times. If the response times are greater than .025 (25 milliseconds), which is a conservative threshold, then noticeable slow-downs and performance issues affecting users may be occurring. For more information, refer to Logical/Physical Disk Read/Write Latency Analysis in this topic.
Disk: Logical Disk Transfers/sec	“Disk Transfers/sec” is the rate of read and write operations on the disk. The thresholds for this analysis check to see whether any of the logical disks are showing poor response times (greater than 25 ms response times for I/O operations). If this is true, then the disk transfers per second should be at or above 80. If not, then the disk architecture needs to be investigated. The most common cause of poor disk I/O is LUN overloading on the SAN. For more information, refer to Logical Disk Transfers/sec in this topic.
Disk: LogicalDisk % Free Space	“% Free Space” is the percentage of total usable space that was free on the selected

Analysis Type and Name	Analysis Description
	<p>logical disk drive. Performance should not be affected until the available disk drive space is less than 30 percent. When 70 percent of the disk drive is used, the remaining free space is located closer to the disk's spindle at the center of the disk drive, which operates at a lower performance level. Lack of free disk space can cause severe disk performance.</p> <p>Reference: NT Server and Disk Subsystem Performance at http://go.microsoft.com/fwlink/?linkid=106783</p>
Disk: Process IO Data Operations/sec and Process IO Other Operations/sec Analysis	<p>These counters count all I/O activity generated by the process to include file, network and device I/Os. These analyses check when processes are doing more than 1,000 I/O's per second and flag it as a warning. These analyses are best used in correlation with other analyses such as disk analysis to determine which processes might be involved in the I/O activity.</p>
Memory: Available Memory	<p>This analysis checks whether the total available memory is low – Warning at 10 percent available and Critical at 5 percent available. A warning is also alerted when a decreasing trend of 10 MB's per hour is detected, which can indicate a potential upcoming memory condition. Low physical memory can cause increased privileged mode CPU and system delays. For more information, refer to Available MemoryAnalysis in this topic.</p>
Memory: Free System Page Table Entries	<p>Free System Page Table Entries (PTE's) are the number of page table entries not currently used by the system. This analysis determines whether the system is running out of PTE's by checking whether there are fewer than 5,000 free PTE's with a Warning if there are fewer than 10,000 free PTE's. Lack of enough PTE's can result in systemwide hangs. Also note that the /3GB switch will lower the amount of free PTE's significantly. For more information, refer</p>

Analysis Type and Name	Analysis Description
	to Free System Page Table Entries Analysis in this topic.
Memory: Memory Leak Detection	<p>This analysis determines whether any of the processes are consuming a large amount of the system's memory and whether the process is increasing in memory consumption over time. A process consuming large portions of memory is fine as long as the process returns the memory back to the system. Look for increasing trends in the chart. An increasing trend over a long period of time could indicate a memory leak. "Private Bytes" is the current size, in bytes, of memory that this process has allocated that cannot be shared with other processes. This analysis checks for 10 MB's per hour and 5 MB's per hour increasing trends. Use this analysis in correlation with the Available Memory analysis in PAL. For more information, refer to Memory Leak Detection Analysis in this topic.</p>
Memory: Handle Leak Detection	<p>This analysis checks all of the processes to determine how many handles each has open and to determine whether a handle leak is suspected. A process with a large number of handles and/or an aggressive upward trend could indicate a handle leak, which typically results in a memory leak. The total number of handles currently open by this process is equal to the sum of the handles currently open by each thread in this process.</p> <p>Reference: Debug Diagnostic Tool v1.1 at http://go.microsoft.com/fwlink/?linkid=106784</p>
Memory: Memory Pages Input/sec	<p>"Pages Input/sec" is the rate at which pages are read from disk to resolve hard page faults. Hard page faults occur when a process refers to a page in virtual memory that is not in its working set or elsewhere in physical memory, and must be retrieved from disk. This analysis checks whether there are more than 10 page file reads per second.</p>

Analysis Type and Name	Analysis Description
Memory: Memory Pages/sec	<p>This analysis checks to see whether the “Pages/sec” is high (above 1,000). If it is high, then the system is likely running out of memory by trying to page the memory to the disk. “Pages/sec” is the rate at which pages are read from or written to disk to resolve hard page faults. This counter is a primary indicator of the kinds of faults that cause systemwide delays. Use this analysis in correlation with Available Memory analysis and Memory Leak Detection analysis in PAL. If all of these analyses are throwing alerts at the same time, then this may indicate the system is running out of memory and the suspected processes involved and follow analysis steps mentioned in the Memory Leak Detection analysis in PAL.</p> <p>For more information, refer to Memory Pages/sec Analysis in this topic.</p>
Memory: Memory System Cache Resident Bytes	<p>“System Cache Resident Bytes” is the size, in bytes, of the pageable operating system code in the file system cache. This value includes only current physical pages and does not include any virtual memory pages not currently resident. This value is a component of “Memory\\System Code Resident Bytes” which represents all pageable operating system code that is currently in physical memory. This counter displays the last observed value only; it is not an average. This analysis checks for an increasing trend of 10 MB’s per hour. Under load, a server might use the system cache in order to cache I/O activity such as disk. Use in correlation with Process IO Data Operations/sec and Process IO Other Operations/sec analyses in PAL.</p> <p>Reference: File Cache Performance and Tuning at http://go.microsoft.com/fwlink/?linkid=106786</p>
Memory: Pool Non Paged Bytes	<p>“Pool Nonpaged Bytes” is the size, in bytes, of the non-paged pool, an area of system memory for objects that cannot be written to disk, but</p>

Analysis Type and Name	Analysis Description
	<p>must remain in physical memory as long as they are allocated. This analysis checks to see whether the system is coming close to the maximum pool non paged memory size. It does this by estimating the pool sizes taking into consideration /3GB, physical memory size, and 32-bit/64-bit, then determining whether the value is higher than 60 percent of the estimated pool size. If the system becomes close to the maximum size, then the system could experience system wide hangs.</p> <p>The /3GB switch option in the boot.ini file significantly reduces the size of this memory pool.</p> <p>For more information, refer to Pool Non-Paged Bytes Analysis in this topic.</p>
Memory: Pool Paged Bytes	<p>This analysis checks to see whether the system is coming close to the maximum pool paged memory size. It does this by estimating the pool sizes taking into consideration /3GB, physical memory size, and 32-bit/64-bit, then determining whether the value is higher than 60 percent of the estimated pool size. If the system becomes close to the maximum size, then the system could experience system wide hangs.</p> <p>The /3GB switch option in the boot.ini file significantly reduces the size of this memory pool.</p> <p>For more information, refer to Pool Paged Bytes Analysis in this topic.</p>
Memory: Process Thread Count	<p>This analysis checks all of the processes to determine whether a process has more than 500 threads and if the number of threads is increasing by 50 threads per hour. A process with a large number of threads and/or an aggressive upward trend could indicate a thread leak which typically results in either a memory leak or high context switching. High context switching will result in high privileged mode CPU.</p>

Analysis Type and Name	Analysis Description
Memory: Process Working Set	<p>“Working Set” is the current size, in bytes, of the working set of this process. The working set is the set of memory pages touched recently by the threads in the process. If free memory in the computer is above a threshold, pages are left in the working set of a process even if they are not in use. When free memory falls below a threshold, pages are trimmed from working sets. If they are needed they will then be soft-faulted back into the working set before leaving main memory. This analysis checks for an increasing trend of 10 MB’s or more in each of the processes. Use in correlation with Available Memory analysis in PAL.</p> <p>Reference: Detecting Memory Bottlenecks at http://go.microsoft.com/fwlink/?linkid=106787</p>
Network: Network Output Queue Length Analysis	<p>This analysis checks to see how many threads are waiting on the network adapter. If a lot of threads are waiting on the network adapter, then the system is probably saturating the network I/O due to network latency or network bandwidth. “Output Queue Length” is the length of the output packet queue (in packets). Delays are indicated if this is longer than two, and the bottleneck should be found and eliminated, if possible. Typical causes of network output queuing include high numbers of small network requests and network latency.</p>
Network: Network Utilization Analysis	<p>“Bytes Total/sec” is the rate at which bytes are sent and received over each network adapter, including framing characters. “Network Interface\Bytes Received/sec” is a sum of “Network Interface\Bytes Received/sec” and “Network Interface\Bytes Sent/sec”. This counter helps you know whether the traffic at your network adapter is saturated and whether you need to add another network adapter. How quickly you can identify a problem depends on the type of network you have as well as whether you share bandwidth with other</p>

Analysis Type and Name	Analysis Description
	<p>applications. This analysis converts “Bytes Total/sec” to bits and compares it to the current bandwidth of the network adapter to calculate network utilization. Next, it checks for utilization above 50 percent.</p> <p>Reference: Measuring .NET Application Performance at http://go.microsoft.com/fwlink/?linkid=106788</p>
Paging File: Paging File % Usage and % Usage Peak	<p>The amount of the page file instance in use in percent. See also “Process\Page File Bytes”. This analysis checks whether the percentage of usage is greater than 70 percent.</p>
Processor: Processor Utilization Analysis and Excessive Processor Use by Processes	<p>This counter is the primary indicator of processor activity and displays the average percentage of busy time observed during the sample interval. It is calculated by monitoring the time that the service is inactive and subtracting that value from 100 percent. This analysis checks for utilization greater than 60 percent on each processor. If so, determine whether it is high user mode CPU or high privileged mode. If high privileged mode CPU is suspected, then see the Privileged Mode CPU analysis in PAL. If a user-mode processor bottleneck is suspected, then consider using a process profiler to analyze the functions causing the high CPU consumption.</p>
Processor: Processor Queue Length	<p>This analysis determines whether the average processor queue length exceeds the number of processors. If so, then this could indicate a processor bottleneck. Use this analysis in correlation with Privileged Mode CPU analysis and Excessive Processor Use by Process analysis in PAL. For detailed information, refer to Processor Queue Length Analysis in this topic.</p>
Processor: Privileged Mode CPU Analysis	<p>This counter indicates the percentage of time a thread runs in privileged mode. When your application calls operating system functions (for</p>

Analysis Type and Name	Analysis Description
	<p>example to perform file or network I/O or to allocate memory), these operating system functions are executed in privileged mode. This analysis checks to see whether privileged mode CPU is consuming more than 30 percent of total CPU. If so, then the CPU consumption is likely caused by another bottleneck other than the processor such as network, memory, or disk I/O. Use in correlation with Processor: % Interrupt Time and Processor: High Context Switching analyses in PAL. For more information, refer to Privileged Mode CPU Analysis in this topic.</p>
Processor: Interrupt Time	<p>“% Interrupt Time” is the time the processor spends receiving and servicing hardware interrupts during sample intervals. This value is an indirect indicator of the activity of devices that generate interrupts, such as the system clock, the mouse, disk drivers, data communication lines, network interface cards and other peripheral devices. These devices normally interrupt the processor when they have completed a task or require attention. Normal thread execution is suspended during interrupts. Most system clocks interrupt the processor every 10 milliseconds, creating a background of interrupt activity. A dramatic increase in this counter indicates potential hardware problems. This analysis checks for “% Interrupt Time” greater than 30 percent. If this occurs, then consider updating devices drivers for hardware that correlates to this alert.</p> <p>Reference: Measuring .NET Application Performance at http://go.microsoft.com/fwlink/?linkid=106788</p>
Processor: High Context Switching	<p>A context switch happens when a higher priority thread preempts a lower priority thread that is currently running or when a high priority thread blocks. High levels of context switching can occur when many threads share the same</p>

Analysis Type and Name	Analysis Description
	<p>priority level. This often indicates that too many threads are competing for the processors on the system. As a general rule, context switching rates of less than 5,000 per second per processor are not worth worrying about. If context switching rates exceed 15,000 per second per processor, then there is a constraint. For more information, refer to High Context Switching Analysis in this topic.</p>
<p>Microsoft BizTalk Server 2010: BizTalk Dehydrating Orchestrations</p>	<p>When many long-running business processes are running at the same time, memory and performance issues are possible. The orchestration engine addresses these issues by "dehydrating" and "rehydrating" orchestration instances. Dehydration is the process of serializing the state of an orchestration into a SQL Server database. Rehydration is the reverse of this process: deserializing the last running state of an orchestration from the database. Dehydration is used to minimize the use of system resources by reducing the number of orchestrations that have to be instantiated in memory at one time. Therefore, dehydrations save memory consumption, but are relatively expensive operations to perform. This analysis checks for dehydrations of 10 or more. If so, BizTalk Server may be running out of memory (either virtual or physical), a high number of orchestrations are waiting on messages, or the dehydration settings are not set properly.</p> <p>Reference: Orchestration Dehydration and Rehydration at http://go.microsoft.com/fwlink/?LinkId=155284</p>
<p>Microsoft BizTalk Server 2010: BizTalk High Database Sessions</p>	<p>This counter has two possible values: normal (0) or exceeded (1). This analysis checks for a value of 1. If so, BizTalk has exceeded the threshold of the number of database sessions permitted. This value is controlled by the "Database connection per CPU" value in the</p>

Analysis Type and Name	Analysis Description
	<p>BizTalk Host Throttling settings. “Database connection per CPU” is the maximum number of concurrent database sessions (per CPU) allowed before throttling begins. You can monitor the number of active Database connections by using the Database session performance counter under the BizTalk:Message Agent performance object category. This parameter only affects outbound message throttling. For more information, refer to BizTalk High Database Sessions Analysis in this topic.</p>
<p>Microsoft BizTalk Server 2010: BizTalk High Database Size</p>	<p>This counter will be set to a value of 1 if either of the conditions listed for the message count in database threshold occurs. By default the host message count in database throttling threshold is set to a value of 50,000, which will trigger a throttling condition under the following circumstances:</p> <ul style="list-style-type: none"> • The total number of messages published by the host instance to the work, state, and suspended queues of the subscribing hosts exceeds 50,000. • The number of messages in the spool table or the tracking table exceeds 500,000 messages. <p>If this occurs, then consider a course of action that will reduce the number of messages in the database. For example, ensure the SQL Server jobs in BizTalk Server are running without error and use the Group Hub page in the BizTalk Server Administration console to determine whether message build up is caused by large numbers of suspended messages. For more information, refer to BizTalk High Database Size Analysis in this topic.</p>
<p>Microsoft BizTalk Server 2010: BizTalk High In-Process Message Count</p>	<p>This analysis checks the High In-Process Message Count counter to determine whether this kind of throttling is occurring. If so, consider adjusting the “In-Process messages per CPU”</p>

Analysis Type and Name	Analysis Description
	<p>setting. This parameter only affects outbound message throttling. Enter a value of 0 in the “In-Process messages per CPU” setting to disable throttling based on the number of in-process messages per CPU. The default value for the “In-Process messages per CPU” setting is 1,000. Note that modifying this value can also have an impact on low latency of messages and/or the efficiency of BizTalk resources. For more information, refer to BizTalk High In-Process Message Count Analysis in this topic.</p>
<p>Microsoft BizTalk Server 2010: BizTalk High Message Delivery Rate</p>	<p>This analysis checks for a value of 1 in the High Message Delivery Rate counter. High message delivery rates can be caused by high processing complexity, slow outbound adapters, or a momentary shortage of system resources. For more information, refer to BizTalk High Message Delivery Rate Analysis in this topic.</p>
<p>Microsoft BizTalk Server 2010: BizTalk High Message Publishing Rate</p>	<p>Inbound host throttling, also known as message publishing throttling in BizTalk Server, is applied to host instances that contain receive adapters or orchestrations that publish messages to the MessageBox database. This analysis checks for a value of 1 in the High Message Publishing Rate counter. If this occurs, then the database cannot keep up with the publishing rate of messages to the BizTalk MessageBox database.</p> <p>References:</p> <ul style="list-style-type: none"> • Host Throttling Performance Counters at http://go.microsoft.com/fwlink/?LinkId=155285 • How BizTalk Server Implements Host Throttling at http://go.microsoft.com/fwlink/?LinkId=155286 • Message Publishing Throttling Settings Dialog Box at http://go.microsoft.com/fwlink/?LinkId=1552

Analysis Type and Name	Analysis Description
	<p>87</p> <ul style="list-style-type: none"> What is Host Throttling? at http://go.microsoft.com/fwlink/?LinkID=154694
Microsoft BizTalk Server 2010: BizTalk High Process Memory	The BizTalk Process Memory usage throttling threshold setting is the percentage of memory used compared to the sum of the working set size and total available virtual memory for the process if a value from 1 through 100 is entered. When a percentage value is specified the process memory threshold is recalculated at regular intervals. This analysis checks for a value of 1 in the High Process Memory counter. For more information, refer to BizTalk High Process Memory Analysis in this topic.
Microsoft BizTalk Server 2010: BizTalk High System Memory	The BizTalk Physical Memory usage throttling threshold setting is the percentage of memory consumption compared to the total amount of available physical memory if a value from 1 through 100 is entered. This setting can also be the total amount of available physical memory in megabytes if a value greater than 100 is entered. Enter a value of 0 to disable throttling based on physical memory usage. The default value is 0. For more information, refer to BizTalk High System Memory Analysis in this topic.
Microsoft BizTalk Server 2010: BizTalk High Thread Count	“Threads Per CPU” is the total number of threads in the host process including threads used by adapters. If this threshold is exceeded, BizTalk Server will try to reduce the size of the EPM thread pool and message agent thread pool. Thread based throttling should be enabled in scenarios where high load can lead to the creation of a large number of threads. This parameter affects both inbound and outbound throttling. Thread based throttling is disabled by default. For more information, refer to BizTalk High Thread Count Analysis in this topic.
Microsoft BizTalk Server 2010: BizTalk Host	The BizTalk Host Queue Length tracks the total

Analysis Type and Name	Analysis Description
Queue Length	<p>number of messages in the particular host queue. You can use length size (i.e., 'BizTalk:MessageBox:HostCounters:Host Queue – Length') to give a more detailed view of the number of messages being queued up internally by showing the queue depth for an individual host. This counter can be useful in determining if a specific host is bottlenecked. Assuming unique hosts are used for each transport, this can be helpful in determining potential transport bottlenecks. This analysis checks for average queue lengths greater than 1. The Host Queue Length is a weighted Queue length by aggregating the record count of all the Queues (Work Q, State Q, Suspended Q) of the target host.</p> <p>Reference: BizTalk Server 2006: Managing a Successful Performance Lab at http://go.microsoft.com/fwlink/?linkid=104152</p> <p> Note This white paper is also applicable to BizTalk Server 2010.</p>
Microsoft BizTalk Server 2010: BizTalk Host Suspended Messages Queue Length	<p>This counter tracks the total number of suspended messages for the particular host. A suspended message is an instance of a message or orchestration that BizTalk Server has stopped processing due to an error in the system or the message. Generally, suspended instances caused by system errors are resumable upon resolution of the system issue. Often, suspended instances due to a message problem are not resumable, and the message itself must be fixed and resubmitted to the BizTalk Server system.</p> <p>The suspended message queue is a queue that contains work items for which an error or failure was encountered during processing. A suspended queue stores the messages until they can be corrected and reprocessed, or deleted. This analysis checks for any</p>

Analysis Type and Name	Analysis Description
	<p>occurrence of suspended messages. An increasing trend could indicate severe processing errors.</p> <p>Reference: BizTalk Server 2004: Monitoring and Troubleshooting at http://go.microsoft.com/fwlink/?linkid=108771</p>
<p>BizTalk Server 2010: BizTalk Idle Orchestrations</p>	<p>Number of idle orchestration instances currently hosted by the host instance. This counter refers to orchestrations that are not making progress but are not dehydratable. This situation can occur when the orchestration is blocked, waiting for a receive, listen, or delay in an atomic transaction. If a large number of non-dehydratable orchestrations accumulate, then BizTalk may run out of memory.</p> <p>Dehydration is the process of serializing the state of an orchestration into a SQL Server database. Rehydration is the reverse of this process: deserializing the last running state of an orchestration from the database.</p> <p>Dehydration is used to minimize the use of system resources by reducing the number of orchestrations that have to be instantiated in memory at one time. The engine dehydrates the instance by saving the state, and frees up the memory required by the instance. By dehydrating dormant orchestration instances, the engine makes it possible for a large number of long-running business processes to run concurrently on the same computer. This analysis checks for an increasing trend of one idle orchestration per hour.</p> <p>Reference: Orchestration Dehydration and Rehydration at http://go.microsoft.com/fwlink/?LinkId=155284</p>
<p>BizTalk Server 2010: BizTalk Inbound Latency</p>	<p>Average latency in milliseconds from when the messaging engine receives a document from the adapter until the time it is published to the MessageBox. Reducing latency is important to some users of BizTalk, therefore tracking how</p>

Analysis Type and Name	Analysis Description
	<p>much time documents spend in the inbound adapter is important. For more information, refer to BizTalk Inbound Latency Analysis in this topic.</p>
<p>BizTalk Server 2010: BizTalk Message Delivery Delay</p>	<p>This is the current delay in milliseconds (ms) imposed on each message delivery batch (applicable if the message delivery is being throttled). In regards to throttling, a delay is applied in the publishing or processing of the message, depending on whether the message is inbound or outbound. The delay period is proportional to the severity of the throttling condition. Higher severity throttling conditions will initiate a longer throttling period than lower severity throttling conditions. This delay period is adjusted up and down within certain ranges by the throttling mechanism as conditions change. The current delay period is exposed through the message delivery delay (ms) and the message publishing delay (ms) performance counters associated with the BizTalk:Message Agent performance object category. This analysis checks for a message delivery delay of greater than 5 seconds. Long message delivery delays may indicate heavy throttling due to high load.</p> <p>Reference: How BizTalk Server Implements Host Throttling at http://go.microsoft.com/fwlink/?LinkId=155286</p>
<p>BizTalk Server 2010: BizTalk Message Delivery Throttling State</p>	<p>The BizTalk message delivery throttling state is one of the primary indicators of throttling. It is a flag indicating whether the system is throttling message delivery (affecting XLANG message processing and outbound transports). The throttling condition is indicated by the numeric value of the counter. For more information, see BizTalk Message Delivery Throttling State Analysis in this topic.</p>
<p>BizTalk Server 2010: BizTalk Message Publishing Delay</p>	<p>The delay injected in each qualifying batch for throttling the publishing of messages. In</p>

Analysis Type and Name	Analysis Description
	<p>regards to throttling, a delay is applied in the publishing or processing of the message, depending on whether the message is inbound or outbound. The delay period is proportional to the severity of the throttling condition. Higher severity throttling conditions will initiate a longer throttling period than lower severity throttling conditions. This delay period is adjusted up and down within certain ranges by the throttling mechanism as conditions change. The current delay period is exposed through the message delivery delay (ms) and the message publishing delay (ms) performance counters associated with the BizTalk:Message Agent performance object category. This analysis checks for a message publishing delay of greater than 5 seconds. Long message delivery delays may indicate heavy throttling due to high load.</p> <p>Reference: How BizTalk Server Implements Host Throttling at http://go.microsoft.com/fwlink/?LinkId=155286</p>
<p>BizTalk Server 2010: BizTalk MessageBox Database Connection Failures</p>	<p>This performance counter is the number of attempted database connections that failed since the host instance started. If the SQL Server service hosting the BizTalk databases becomes unavailable for any reason, the database cluster transfers resources from the active computer to the passive computer. During this failover process, the BizTalk Server service instances experience database connection failures and automatically restart to reconnect to the databases. The functioning database computer (previously the passive computer) begins processing the database connections after assuming the resources during failover. For more information, refer to BizTalk MessageBox Database Connection Failures Analysis in this topic.</p>
<p>BizTalk Server 2010: BizTalk Messaging Latency: Request Response Latency</p>	<p>Average latency in milliseconds from when the Messaging Engine receives a request</p>

Analysis Type and Name	Analysis Description
	<p>document from the adapter until the time a response document is given back to the adapter. Refer to the chart showing how latency is measured in BizTalk Inbound Latency Analysis in this topic. Assuming a low latency environment, this analysis checks for a Request-Response Latency greater than 5 seconds. This may indicate a processing delay between the inbound adapter and the outbound adapter.</p> <p>References:</p> <ul style="list-style-type: none"> Request/Response Messaging at http://go.microsoft.com/fwlink/?LinkId=155288 BizTalk Server 2006: Scalability Case Study Using the SOAP Adapter in BizTalk Server 2006 at http://go.microsoft.com/fwlink/?linkid=108774
<p>BizTalk Server 2010: BizTalk Messaging Publishing Throttling State</p>	<p>The BizTalk message publishing throttling state is one of the primary indicators of throttling. It is a flag indicating whether the system is throttling message publishing (affecting XLANG message processing and inbound transports). For more information, refer to BizTalk Messaging Publishing Throttling State Analysis in this topic.</p>
<p>BizTalk Server 2010: BizTalk Orchestration Suspended/second</p>	<p>A suspended message is an instance of a message or orchestration that BizTalk Server has stopped processing due to an error in the system or the message. Generally, suspended instances caused by system errors are resumable upon resolution of the system issue. Often, suspended instances due to a message problem are not resumable, and the message itself must be fixed and resubmitted to the BizTalk Server system. This analysis checks for any suspended messages/orchestrations.</p> <p>Reference: BizTalk Server 2004: Monitoring and Troubleshooting at http://go.microsoft.com/fwlink/?linkid=108771</p>

Analysis Type and Name	Analysis Description
BizTalk Server 2010: BizTalk Orchestrations Completed/second	<p>This is the number of BizTalk orchestrations that have completed per second. This is a good indicator as to how much throughput BizTalk is processing. This analysis provides statistics only.</p> <p>Reference: BizTalk Server 2006: Scalability Case Study Using the SOAP Adapter in BizTalk Server 2006 http://go.microsoft.com/fwlink/?linkid=108774</p>
BizTalk Server 2010: BizTalk Orchestrations Discarded	<p>Number of orchestration instances discarded from memory since the host instance started. An orchestration can be discarded if the engine fails to persist its state. This analysis checks for any discarded messages.</p> <p>Reference: BizTalk Core Engine's WebLog http://go.microsoft.com/fwlink/?linkid=108775</p>
BizTalk Server 2010: BizTalk Orchestrations Resident in Memory	<p>Number of orchestration instances currently hosted by the host instance. This analysis checks for an increasing trend in orchestrations resident in memory and whether more than 50 percent of the orchestrations resident in memory are not dehydratable. For more information, refer to BizTalk Orchestrations Resident in Memory Analysis.</p>
BizTalk Server 2010: BizTalk Outbound Adapter Latency	<p>This is the average latency in seconds from when the adapter gets a document from the messaging engine until the time it is sent by the adapter. Refer to the chart showing how latency is measured in BizTalk Inbound Latency Analysis in this topic. Assuming a low latency environment, this analysis checks for latency in the outbound adapter of greater than 5 seconds on average. This may indicate a processing delay in the transport of messages through outbound adapters in this host instance. If multiple outbound adapters exist in this host instance, then consider separating them into their own hosts to determine which outbound adapter has high latency.</p>

Analysis Type and Name	Analysis Description
	<p>References:</p> <ul style="list-style-type: none"> • Request/Response Messaging at http://go.microsoft.com/fwlink/?LinkId=155288 • BizTalk Server 2006: Scalability Case Study Using the SOAP Adapter in BizTalk Server 2006 at http://go.microsoft.com/fwlink/?linkid=108774 • Identifying Bottlenecks in the BizTalk Tier at http://go.microsoft.com/fwlink/?LinkId=155289 • BizTalk Server 2004: Performance Tuning for Low Latency Messaging at http://go.microsoft.com/fwlink/?linkid=108777
<p>BizTalk Server 2010: BizTalk Pending Messages</p>	<p>The number of received messages that have not been acknowledged as received to the MessageBox. Pending messages are messages that have been pulled into memory and delivered to the XLANG orchestration, but have not yet been processed. This circumstance has nothing to do with data loss. Delivering a message to an orchestration is a multi-step process and is simply an instance of the message residing in the spool table in the database. These pending messages count as in-process messages; therefore, having a large number of them in memory could cause memory throttling on the system. Adjusting the Internal Message Queue Size setting could help with controlling the number of pending messages. The In-Process Messages Per CPU setting has an impact on when throttling will invoke when a high number of pending messages occurs. These setting are found in the Host properties in the BizTalk Administration Console. This analysis checks only shows statistics for this counter.</p> <p>Reference: Orchestration Engine Performance Counters at</p>

Analysis Type and Name	Analysis Description
	http://go.microsoft.com/fwlink/?LinkId=155290
BizTalk Server 2010: BizTalk Persistence Points/second	<p>Average number of orchestration instances persisted per second. The orchestration engine saves the state of a running orchestration instance at various points. If it needs to rehydrate the orchestration instance, start up from a controlled shutdown, or recover from an unexpected shutdown, it will run the orchestration instance from the last persistence point. In order to persist an orchestration instance, all object instances that your orchestration refers to directly or indirectly (as through other objects) must be serializable. As message-persistence frequency (the number of times that data needs to be persisted) increases, overall performance decreases. In effect, each persistence point is a round trip to the database, so whenever possible reduce the frequency of persistence points by avoiding or consolidating persistence points. See the references below for more information regarding when persistence points occur. This analysis checks for more than 10 persistence points per second on average. This is a general starting point.</p> <p>References:</p> <ul style="list-style-type: none"> • Persistence in Orchestrations at http://go.microsoft.com/fwlink/?LinkId=155291 • Persistence and the Orchestration Engine at http://go.microsoft.com/fwlink/?LinkId=155292
BizTalk Server 2010: BizTalk Private Bytes	<p>This is the megabytes of allocated private memory for the host instance and comparable to the “Process(*)\Private Bytes” performance counter. This analysis determines whether any of the host instances are consuming a large size of the system's memory and whether the host instance is increasing in memory</p>

Analysis Type and Name	Analysis Description
	consumption over time. Refer to BizTalk Private Bytes Analysis in this topic for more information.
BizTalk Server 2010: BizTalk Spool Table Size	<p>The MessageBox spool table contains a record for each message in the system (active or waiting to be "garbage collected"). Monitoring the number of rows in this table and the number of messages received per second while increasing system load provides an easy way to find the maximum sustainable throughput. Simply increase the input load until either 1) the spool table starts to grow indefinitely or 2) the number of messages received per second plateaus, whichever comes first, and that is your maximum sustainable throughput. In summary, regardless of other indicators, this measure will give you a quick and easy way to assess whether your system is being overdriven or not. When the BizTalk spool tables size is on an increasing trend, then throttling due to imbalanced message delivery rate (input rate exceeds output rate) or throttling due to Database size may occur. This analysis checks for an increasing trend in the BizTalk Spool Table Size.</p> <p>References:</p> <ul style="list-style-type: none"> • Understanding BizTalk Server 2004 SP1 Throughput and Capacity at http://go.microsoft.com/fwlink/?linkid=108781 • Sustainable Load Test at http://go.microsoft.com/fwlink/?LinkId=155293 • Recommendations When Testing Engine Performance at http://go.microsoft.com/fwlink/?LinkID=155294
BizTalk Server 2010: BizTalk Tracking Data Size	As BizTalk Server processes more and more data on your system, the BizTalk Tracking database (BizTalkDTADb) continues to grow in size. Unchecked growth decreases system

Analysis Type and Name	Analysis Description
	<p>performance and may generate errors in the Tracking Data Delivery Service (TDDS). In addition to general tracking data, tracked messages can also accumulate in the MessageBox database, causing poor disk performance. This analysis checks for an increasing trend of more than 5 MB's per hour in the tracking data size.</p> <p>Reference: Archiving and Purging the BizTalk Tracking Database at http://go.microsoft.com/fwlink/?LinkID=153816</p>
<p>BizTalk Server 2010: BizTalk Transactional Scopes Aborted</p>	<p>This is the number of long-running or atomic scopes that have been aborted since the host instance started. A transactional scope abort is a failure that occurs in a transaction scope within an orchestration. It is important to understand that the compensation handler of a scope is invoked only if the scope completed successfully, but then is required to be undone because a surrounding scope has decided to abort (due to failures that may occur later in the process). Also, no "auto" rollback of state occurs in case of a transaction abort. You can achieve this outcome programmatically through the exception and compensation handlers. Transactional scope aborts should not normally occur in a production environment; therefore, this analysis checks for the occurrence of any transactional scopes aborted.</p> <p>Reference: Transactions Across BizTalk Server 2004 at http://go.microsoft.com/fwlink/?linkid=108784</p>
<p>BizTalk Server 2010: BizTalk Transactional Scopes Compensated</p>	<p>Compensation can be thought of as a logical undo of the work that has been successfully committed in response to some error condition. It is important to understand that the compensation handler of a scope is invoked only if the scope completed successfully, but then is required to be undone because a</p>

Analysis Type and Name	Analysis Description
	<p>surrounding scope has decided to abort (due to failures that may occur later in the process). Also, no "auto" rollback of state occurs in case of a transaction abort. You can achieve this programmatically through the exception and compensation handlers. Transactional scope compensations should not normally occur in a production environment; therefore, this analysis checks for the occurrence of any transactional scopes aborted.</p> <p>Reference: Transactions Across BizTalk Server 2004 at http://go.microsoft.com/fwlink/?linkid=108784</p>
BizTalk Server 2010: BizTalk Virtual Bytes	<p>This is the megabytes reserved for virtual memory for the host instance. This analysis determines whether any of the host instances are consuming a large amount of the system's memory and whether the host instance is increasing in memory consumption over time. For more information, refer to BizTalk Virtual Bytes Analysis in this topic.</p>
BizTalk Server 2010: BizTalk Message Agent Database Session Throttling	<p>This is the number of open database connections to the MessageBox compared to its respective BizTalk throttling setting. "Database connection per CPU" is the maximum number of concurrent database sessions (per CPU) allowed before throttling begins. For more information, refer to BizTalk Message Agent Database Session Throttling Analysis in this topic.</p>
BizTalk Server 2010: BizTalk Message Agent Database Session Throttling Threshold	<p>This is the current threshold for the number of open database connections to the MessageBox. "Database connection per CPU" is the maximum number of concurrent database sessions (per CPU) allowed before throttling begins. For more information, refer to BizTalk Message Agent Database Session Throttling Threshold Analysis in this topic.</p>
BizTalk Server 2010: BizTalk Message Agent	<p>This is the number of concurrent messages that</p>

Analysis Type and Name	Analysis Description
In-process Message Count Throttling	the service class is processing. The “In-process messages per CPU” setting in the Host Throttling Settings is the maximum number of messages delivered to the End Point Manager (EPM) or XLANG that have not been processed. For more information, refer to BizTalk Message Agent In-process Message Count Throttling Analysis in this topic.
BizTalk Server 2010: BizTalk Message Agent In-process Message Count Throttling Threshold	This is the current threshold for the number of concurrent messages that the service class is processing. The “In-process messages per CPU” setting in the Host Throttling Settings is the maximum number of messages delivered to the End Point Manager (EPM) or XLANG that have not been processed. For more information, refer to BizTalk Message Agent In-process Message Count Throttling Threshold Analysis in this topic.
BizTalk Server 2010: BizTalk Message Agent Process Memory Usage (MB) Throttling	This is the memory usage of current process (MB). BizTalk process memory throttling can occur if the batch to be published has steep memory requirements, or if too many threads are processing messages. For more information, refer to BizTalk Message Agent Process Memory Usage (MB) Throttling Analysis in this topic.
BizTalk Server 2010: BizTalk Message Agent Process Memory Usage (MB) Throttling Threshold	This is the current threshold for the memory usage of current process (MB). The threshold may be dynamically adjusted depending on the actual amount of memory available to this process and its memory consumption pattern. BizTalk process memory throttling can occur if the batch to be published has steep memory requirements, or if too many threads are processing messages. For more information, refer to BizTalk Message Agent Process memory usage (MB) Throttling Threshold Analysis in this topic.
BizTalk Server 2010: BizTalk Message Agent	The total number of threads in the BizTalk process. “Threads Per CPU” is the total number

Analysis Type and Name	Analysis Description
Thread Count Throttling	<p>of threads in the host process including threads used by adapters. If this threshold is exceeded, BizTalk Server will try to reduce the size of EPM thread pool and message agent thread pool. Thread based throttling should be enabled in scenarios where high load can lead to the creation of a large number of threads. This parameter affects both inbound and outbound throttling. Thread based throttling is disabled by default. This analysis checks whether the BizTalk Thread count is greater than 80 percent of the throttling threshold value indicating a throttling condition is likely.</p> <p>References:</p> <ul style="list-style-type: none"> • Host Throttling Performance Counters at http://go.microsoft.com/fwlink/?LinkId=155285 • How BizTalk Server Implements Host Throttling at http://go.microsoft.com/fwlink/?LinkId=155286 • How to Modify the Default Host Throttling Settings at http://go.microsoft.com/fwlink/?LinkId=155296 • Configuration Parameters that Affect Adapter Performance at http://go.microsoft.com/fwlink/?LinkId=154200 • Threads, DB sessions, and throttling at http://go.microsoft.com/fwlink/?linkid=106793
BizTalk Server 2010: BizTalk Message Agent Thread Count Throttling Threshold	<p>This is the current threshold for the total number of threads in the process. "Threads Per CPU" is the total number of threads in the host process including threads used by adapters. If this threshold is exceeded, BizTalk Server will try to reduce the size of EPM thread pool and message agent thread pool. Thread-based throttling should be enabled in scenarios where high load can lead to the creation of a large</p>

Analysis Type and Name	Analysis Description
	<p>number of threads. This parameter affects both inbound and outbound throttling.</p> <p>This analysis checks whether this throttling setting is set to a non-default value. Thread based throttling is disabled by default.</p> <p>References:</p> <ul style="list-style-type: none"> • Host Throttling Performance Counters at http://go.microsoft.com/fwlink/?LinkId=155285 • How BizTalk Server Implements Host Throttling at http://go.microsoft.com/fwlink/?LinkId=155286 • How to Modify the Default Host Throttling Settings at http://go.microsoft.com/fwlink/?LinkId=155296 • Configuration Parameters that Affect Adapter Performance at http://go.microsoft.com/fwlink/?LinkID=154200 • Threads, DB sessions, and throttling at http://go.microsoft.com/fwlink/?linkid=106793

Logical/Physical Disk Read/Write Latency Analysis

The most reliable way for Windows to detect a disk performance bottleneck is by measuring its response times. If the response times are greater than .025 (25 milliseconds), which is a conservative threshold, then noticeable slow-downs and performance issues affecting users may be occurring.

Common causes of poor disk latency are disk fragmentation, performance cache, an over saturated SAN, and too much load on the disk. Use the SPA tool to help identify the top files and processes using the disk. Also check the “Process IO Data Operations/sec” and “Process IO Other Operations/sec” to see which processes are consuming the most disk I/O’s. Keep in mind that performance monitor counters are unable to specify which files are involved.

References

- Ruling Out Disk-Bound Problems at <http://go.microsoft.com/fwlink/?linkid=50669>
- How to Identify a Disk Performance Bottleneck Using the Microsoft Server Performance Advisor (SPA) Tool at <http://go.microsoft.com/fwlink/?linkid=98096>

- Download Details for Microsoft Service Performance Advisor (SPA) at: <http://go.microsoft.com/fwlink/?linkid=57769>

Logical Disk Transfers/sec

“Disk Transfers/sec” is the rate of read and write operations on the disk. While disk transfers are not a direct correlation to disk I/O's, they do tell us how many disk operations are occurring. If you average out sequential I/O's and random I/O's, then you end up with about 80 I/O's per second as a general rule of thumb. Therefore, we should expect a SAN drive to perform more than 80 I/O's per second when under load. The thresholds for this analysis check to see whether any of the logical disks are showing poor response times (greater than 25 ms response times for I/O operations). If this is true, then we should expect the disk transfers per second to be at or above 80. If not, then the disk architecture needs to be investigated. The most common cause of poor disk I/O is logical unit number (LUN) overloading on the SAN – meaning the condition where more than one LUN is using the small physical disk array.

Reference

NT Server and Disk Subsystem Performance at <http://go.microsoft.com/fwlink/?linkid=106783>

Available Memory Analysis

“Available Mbytes” is the amount of physical memory available to processes running on the computer, in megabytes. The Virtual Memory Manager continually adjusts the space used in physical memory and on disk to maintain a minimum number of available bytes for the operating system and processes. When available bytes are plentiful, the Virtual Memory Manager lets the working sets of processes grow, or keeps them stable by removing an old page for each new page added. When available bytes are few, the Virtual Memory Manager must trim the working sets of processes to maintain the minimum required.

This analysis checks to see whether the total available memory is low – Warning at 10 percent available and Critical at 5 percent available. A warning is also alerted when a decreasing trend of 10 MB's per hour is detected, indicating a potential upcoming memory condition. Low physical memory can cause increased privileged mode CPU and system delays.

References

- Detecting Memory Bottlenecks at <http://go.microsoft.com/fwlink/?linkid=106787>
- Ruling Out Memory-Bound Problems at <http://go.microsoft.com/fwlink/?linkid=62575>

Memory Leak Detection Analysis

This analysis determines whether any of the processes are consuming a large amount of the system's memory and whether the process is increasing in memory consumption over time. A process consuming large portions of memory is okay as long as the process returns the memory back to the system. Look for increasing trends in the chart. An increasing trend over a long period of time could indicate a memory leak. Private Bytes is the current size, in bytes, of memory that this process has allocated that cannot be shared with other processes. This analysis checks for

10 MB's per hour and 5 MB's per hour increasing trends. Use this analysis in correlation with the Available Memory analysis.

Also, keep in mind that newly started processes will initially appear as a memory leak when it is simply normal startup behavior. A memory leak occurs when a process continues to consume memory and does not release memory over a long period of time.

If you suspect a memory leak condition, then install and use the Debug Diag tool. For more information on the Debug Diag Tool, see the references section.

Reference

Debug Diagnostic Tool v1.1 at <http://go.microsoft.com/fwlink/?linkid=106784>

Memory Pages/sec Analysis

This analysis checks to see whether the "Pages/sec" is high. If it is high, then the system is likely running out of memory by trying to page the memory to the disk. "Pages/sec" is the rate at which pages are read from or written to disk to resolve hard page faults. This counter is a primary indicator of the kinds of faults that cause system-wide delays. It is the sum of "Memory\Pages Input/sec" and "Memory\Pages Output/sec". It is counted in numbers of pages, so it can be compared to other counts of pages, such as "Memory\Page Faults/sec".

This counter should always be below 1,000. This analysis checks for values above 1,000. Use this analysis in correlation with Available Memory Analysis and Memory Leak Detection analysis. If all analyses are throwing alerts at the same time, then this may indicate the system is running out of memory. Follow analysis steps mentioned in Additional Information Regarding Memory Leak Detection analysis in this topic.

Reference

Ruling Out Memory-Bound Problems at <http://go.microsoft.com/fwlink/?linkid=62575>

Memory System Cache Resident Bytes Analysis

"System Cache Resident Bytes" is the size, in bytes, of the pageable operating system code in the file system cache. This value includes only current physical pages and does not include any virtual memory pages not currently resident. It does not equal the System Cache value shown in Task Manager. As a result, this value may be smaller than the actual amount of virtual memory in use by the file system cache. This value is a component of "Memory\System Code Resident Bytes", which represents all pageable operating system code that is currently in physical memory. This counter displays the last observed value only; it is not an average.

This analysis checks for an increasing trend of 10 MB's per hour. Under load, a server might use the system cache in order to cache I/O activity such as disk. Use in correlation with Process IO Data Operations/sec and Process IO Other Operations/sec analyses.

Processor Utilization Analysis and Excessive Processor Use by Processes

“% Processor Time” is the percentage of elapsed time that the processor spends to execute a non-idle thread. It is calculated by measuring the duration the idle thread is active in the sample interval, and then subtracting that time from the interval duration. (Each processor has an idle thread that consumes cycles when no other threads are ready to run.) This counter is the primary indicator of processor activity and displays the average percentage of busy time observed during the sample interval. It is calculated by monitoring the time that the service is inactive, and subtracting that value from 100 percent.

This analysis checks for utilization greater than 60 percent on each individual processor. If so, determine whether it is high user mode CPU or high privileged mode. If high privileged mode CPU is suspected, then see the “Privileged Mode CPU Analysis”. If a user-mode processor bottleneck is suspected, then consider using a process profiler to analyze the functions causing the high CPU consumption. See “How To: Identify Functions causing a High User-mode CPU Bottleneck for Server Applications in a Production Environment” article in the references section for more information.

Processor Queue Length Analysis

“Processor Queue Length” is the number of threads in the processor queue. Unlike the disk counters, this counter shows ready threads only, not threads that are running. There is a single queue for processor time even on computers with multiple processors. Therefore, if a computer has multiple processors, you need to divide this value by the number of processors servicing the workload. A sustained processor queue of less than 10 threads per processor is normally acceptable, dependent of the workload.

This analysis determines whether the average processor queue length exceeds the number of processors. If so, then this could indicate a processor bottleneck. Use this analysis in correlation with Privileged Mode CPU Analysis and Excessive Processor Use by Process. The processor queue is the collection of threads that are ready but not able to be executed by the processor because another active thread is currently executing. A sustained or recurring queue of more threads than number of processors is a good indication of a processor bottleneck.

You can use this counter in conjunction with the “Processor\% Processor Time” counter to determine whether your application can benefit from more CPUs. There is a single queue for processor time, even on multiprocessor computers. Therefore, in a multiprocessor computer, divide the “Processor Queue Length” (PQL) value by the number of processors servicing the workload

If the CPU is very busy (90 percent and higher utilization) and the PQL average is consistently higher than the number of processors, then you may have a processor bottleneck that could benefit from additional CPUs. Or, you could reduce the number of threads and queue at the application level. This will cause less context switching, which is good for reducing CPU load. The common reason for a high PQL with low CPU utilization is that requests for processor time arrive randomly, and threads demand irregular amounts of time from the processor. This means that the processor is not a bottleneck. Instead, your threading logic that needs to be improved.

If a user-mode processor bottleneck is suspected, then consider using a process profiler to analyze the functions causing the high CPU consumption. See the “How To: Identify Functions causing a High User-mode CPU Bottleneck for Server Applications in a Production Environment” article in the references section for more information.

Privileged Mode CPU Analysis

This counter indicates the percentage of time a thread runs in privileged mode. When your application calls operating system functions (for example to perform file or network I/O or to allocate memory), these operating system functions are executed in privileged mode.

High privileged mode CPU indicates that the computer is spending too much time in system I/O versus real (user mode) work. “% Privileged Time” is the percentage of elapsed time that the process threads spent executing code in privileged mode. When a Windows system service is called, the service will often run in privileged mode to gain access to system-private data. Such data is protected from access by threads executing in user mode. Calls to the system can be explicit or implicit, such as page faults or interrupts. Unlike some early operating systems, Windows uses process boundaries for subsystem protection in addition to the traditional protection of user and privileged modes. Some work done by Windows on behalf of the application might appear in other subsystem processes in addition to the privileged time in the process.

This analysis checks to see whether privileged mode CPU is consuming more than 30 percent of total CPU. If so, then the CPU consumption is likely caused by another bottleneck other than the processor such as network, memory, or disk I/O. Use in correlation with % Interrupt Time and High Context Switching analyses.

High Context Switching Analysis

A context switch happens when a higher priority thread preempts a lower priority thread that is currently running or when a high priority thread blocks. High levels of context switching can occur when many threads share the same priority level. This often indicates that too many threads are competing for the processors on the system. If you do not see much processor utilization and you see very low levels of context switching, it could indicate that threads are blocked.

High context switching should only be investigated when privileged mode CPU and overall CPU is high. As a general rule, context switching rates of less than 5,000 per second per processor are not worth worrying about. If context switching rates exceed 15,000 per second per processor, then there is a constraint.

This analysis checks for high CPU, high privileged mode CPU, and high (greater than 5,000 per processor) system context switches per second all occurring at the same time. If high context switching is occurring, then reduce the number of threads and processes running on the system.

BizTalk High Database Sessions Analysis

This counter has two possible values namely normal (0) or exceeded (1). This analysis checks for a value of 1. If so, BizTalk has exceeded the threshold of the number of database sessions

permitted. This value is controlled by the “Database connection per CPU” value in the BizTalk Host Throttling settings.

“Database connection per CPU” is the maximum number of concurrent database sessions (per CPU) allowed before throttling begins. The idle database sessions in the common per-host session pool do not add to this count, and this check is made strictly on the number of sessions actually being used by the host instance. This option is disabled by default; typically, this setting should only be enabled if the database server is a bottleneck or for low-end database servers in the BizTalk Server system. You can monitor the number of active database connections by using the database session performance counter under the BizTalk:Message Agent performance object category. This parameter only affects outbound message throttling. Enter a value of 0 to disable throttling that is based on the number of database sessions. The default value is 0.



Note

The “MaxWorkerThreads” registry key influences the number threads available to BizTalk and may help if most of the BizTalk threads are busy with database connections.

References

- Host Throttling Performance Counters at <http://go.microsoft.com/fwlink/?LinkID=155285>
- Threads, DB sessions, and throttling at <http://go.microsoft.com/fwlink/?linkid=106793>
- Configuration Parameters that Affect Adapter Performance at <http://go.microsoft.com/fwlink/?LinkID=154200>
- How to Modify the Default Host Throttling Settings at <http://go.microsoft.com/fwlink/?LinkID=155296>

BizTalk High Database Size Analysis

This counter refers to the number of messages in the database queues that this process has published. This value is measured by the number of items in the queue tables for all hosts and the number of items in the spool and tracking tables. Queue includes the work queue, the state queue and the suspended queue. If a process is publishing to multiple queues, this counter reflects the weighted average of all the queues.

If the host is restarted, statistics held in memory are lost. Since some overhead is involved, BizTalk Server will resume gathering statistics only when there are at least 100 publishes, with 5 percent of the total publishes within the restarted host process.

This counter will be set to a value of 1 if either of the conditions listed for the message count in database threshold occurs. This threshold is documented in the topic How to Modify the Default Host Throttling Settings referenced below. By default the host message count in database throttling threshold is set to a value of 50,000, which will trigger a throttling condition under the following circumstances:

- The total number of messages published by the host instance to the work, state, and suspended queues of the subscribing hosts exceeds 50,000.
- The number of messages in the spool table or the tracking table exceeds 500,000 messages.

Since suspended messages are included in the message count in database calculation, throttling of message publishing can occur even if the BizTalk server is experiencing low or no load.

This analysis checks for a value of 1. If this occurs, then consider a course of action that will reduce the number of messages in the database. For example, ensure the BizTalk SQL Server jobs are running without error and use the Group Hub in the BizTalk Administration console to determine whether message build up is caused by large numbers of suspended messages.

References

- Suspended Messages are Included in the Message Count in Database Throttling Threshold at <http://go.microsoft.com/fwlink/?LinkId=155304>
- How to Modify the Default Host Throttling Settings at <http://go.microsoft.com/fwlink/?LinkID=155296>
- Host Throttling Performance Counters at <http://go.microsoft.com/fwlink/?LinkID=155285>

BizTalk High In-Process Message Count Analysis

The “In-process messages per CPU” setting in the Host Throttling Settings is the maximum number of messages delivered to the End Point Manager (EPM) or XLANG that have not been processed. This number does not include the messages retrieved from database but still waiting for delivery in the in-memory queue. You can monitor the number of in-Process Messages by using the In-process message count performance counter under the BizTalk:Message Agent performance object category. This parameter provides a hint to the throttling mechanism when considering throttling conditions. The actual threshold is subject to self-tuning. You can verify the actual threshold by monitoring the in-process message count performance counter.

This parameter can be set to a smaller value for large message scenarios, where either the average message size is high, or the processing of messages may require a large number of messages. This would be evident if a scenario experiences memory-based throttling too often and if the memory threshold gets auto-adjusted to a substantially low value. Such behavior would indicate that the outbound transport should process fewer messages concurrently to avoid excessive memory usage. Also, for scenarios where the adapter is more efficient when processing a few messages at a time (for example, when sending to a server that limits concurrent connections), this parameter may be tuned to a lower value than the default.

This analysis checks the High In-Process Message Count counter to determine whether this kind of throttling is occurring. If so, consider adjusting the “In-Process messages per CPU” setting. This parameter only affects outbound message throttling. Enter a value of 0 in the “In-Process messages per CPU” setting to disable throttling based on the number of in-process messages per CPU. The default value for the “In-Process messages per CPU” setting is 1,000. Note that modifying this value can also have an impact on low latency of messages and/or the efficiency of BizTalk resources.

References

- How to Modify the Default Host Throttling Settings at <http://go.microsoft.com/fwlink/?LinkID=155296>

- Host Throttling Performance Counters at <http://go.microsoft.com/fwlink/?LinkID=155285>

BizTalk High Message Delivery Rate Analysis

For outbound (delivered) messages, BizTalk Server throttles delivery of messages if the message delivery incoming rate for the host instance exceeds the message delivery outgoing rate * the specified rate overdrive factor (percent) value. The rate overdrive factor (percent) parameter is configurable on the Message Processing Throttling Settings dialog box. Rate-based throttling for outbound messages is accomplished primarily by inducing a delay before removing the messages from the in-memory queue and delivering the messages to the End Point Manager (EPM) or orchestration engine for processing. No other action is taken to accomplish rate-based throttling for outbound messages.

Outbound throttling can cause delayed message delivery and messages may build up in the in-memory queue and cause de-queue threads to be blocked until the throttling condition is mitigated. When de-queue threads are blocked, no additional messages are pulled from the MessageBox into the in-memory queue for outbound delivery.

This analysis checks for a value of 1 in the High Message Delivery Rate counter. High message delivery rates can be caused by high processing complexity, slow outbound adapters, or a momentary shortage of system resources.

References

- Host Throttling Performance Counters at <http://go.microsoft.com/fwlink/?LinkID=155285>
- How BizTalk Server Implements Host Throttling at <http://go.microsoft.com/fwlink/?LinkID=155286>
- How to Modify the Default Host Throttling Settings at <http://go.microsoft.com/fwlink/?LinkID=155296>

BizTalk High Process Memory Analysis

The BizTalk Process Memory usage throttling threshold setting is the percentage of memory used compared to the sum of the working set size and total available virtual memory for the process if a value from 1 through 100 is entered. By default, the BizTalk Process Memory Usage throttling setting is 25. When a percentage value is specified the process memory threshold is recalculated at regular intervals. If the user specifies a percentage value, it is computed based on the available memory to commit and the current Process Memory usage.

This analysis checks for a value of 1 in the High Process Memory counter. If this occurs, then try to determine the cause of the memory increase by using Debug Diag (see references in Memory Leak Detection analysis). Note that it is normal for processes to consume a large portion of memory during startup and this may initially appear as a memory leak, but a true memory leak occurs when a process fails to release memory that it no longer needs, thereby reducing the amount of available memory over time. See the “How to Capture a Memory Dump of a Process that is Leaking Memory” reference below and/or the “Memory Leak Detection” analysis in PAL for more information on how to generically analyze process memory leaks in BizTalk.

High process memory throttling can occur if the batch to be published has steep memory requirements, or too many threads are processing messages. If the system appears to be over-throttling, consider increasing the value associated with the process memory usage threshold for the host and verify that the host instance does not generate an "out of memory" error. If an "out of memory" error is raised by increasing the process memory usage threshold, then consider reducing the values for the internal message queue size and In-process messages per CPU thresholds. This strategy is particularly relevant in large message processing scenarios. In addition, this value should be set to a low value for scenarios having large memory requirement per message. Setting a low value will kick in throttling early on and prevent a memory explosion within the process.

If your BizTalk server regularly runs out of virtual memory, then consider BizTalk Server 64-bit. Each process on 64-bit servers can address up to 4 TB's of virtual memory versus the 2 GB's in 32-bit. In general, 64-bit BizTalk and 64-bit SQL Server is highly recommended. See the "BizTalk Server 64-bit Support" reference for more information.

References

- How BizTalk Server Implements Host Throttling at <http://go.microsoft.com/fwlink/?LinkID=155286>
- How to Modify the Default Host Throttling Settings at <http://go.microsoft.com/fwlink/?LinkID=155296>
- How to Capture a Memory Dump of a Process that is Leaking Memory at <http://go.microsoft.com/fwlink/?LinkId=155305>
- BizTalk Server 64-bit Support at <http://go.microsoft.com/fwlink/?LinkId=155306>

BizTalk High System Memory Analysis

The BizTalk Physical Memory usage throttling threshold setting is the percentage of memory consumption compared to the total amount of available physical memory if a value from 1 through 100 is entered. This setting can also be the total amount of available physical memory in megabytes if a value greater than 100 is entered. Enter a value of 0 to disable throttling based on physical memory usage. The default value is 0.

This analysis checks for a value of 1 in the High System Memory counter. Since this measures total system memory, a throttling condition may be triggered if non-BizTalk Server processes are consuming an extensive amount of system memory. Therefore if this occurs, the best approach is to identify which processes are consuming the most physical memory and/or add additional physical memory to the server. Also, consider reducing load by reducing the default size of the EPM thread pool, and/or the size of adapter batches. For more information, see the Memory Leak Detection analysis in PAL.

References

- Host Throttling Performance Counters at <http://go.microsoft.com/fwlink/?LinkID=155285>
- How to Modify the Default Host Throttling Settings at <http://go.microsoft.com/fwlink/?LinkID=155296>

- How BizTalk Server Implements Host Throttling at <http://go.microsoft.com/fwlink/?LinkID=155286>

BizTalk High Thread Count Analysis

“Threads Per CPU” is the total number of threads in the host process including threads used by adapters. If this threshold is exceeded, BizTalk Server will try to reduce the size of the EPM thread pool and message agent thread pool. Thread based throttling should be enabled in scenarios where high load can lead to the creation of a large number of threads. This parameter affects both inbound and outbound throttling. Thread based throttling is disabled by default.



Note

The user-specified value is used as a guideline, and the host may dynamically self-tune this threshold value based on the memory usage patterns and thread requirements of the process.

This analysis checks for a value of 1 in the High Thread Count counter. Consider adjusting the different thread pool sizes to ensure that the system does not create a large number of threads. This analysis can be correlated with Context Switches per Second analysis to determine whether the operating system is saturated with too many threads, but in most cases high thread counts cause more contention on the backend database than on the BizTalk server. For more information about modifying the thread pool sizes see How to Modify the Default Host Throttling Settings in references.

References

- Host Throttling Performance Counters at <http://go.microsoft.com/fwlink/?LinkID=155285>
- How BizTalk Server Implements Host Throttling at <http://go.microsoft.com/fwlink/?LinkID=155286>
- How to Modify the Default Host Throttling Settings at <http://go.microsoft.com/fwlink/?LinkID=155296>
- Configuration Parameters that Affect Adapter Performance at <http://go.microsoft.com/fwlink/?LinkID=154200>
- Threads, DB sessions, and throttling at <http://go.microsoft.com/fwlink/?linkid=106793>

BizTalk Inbound Latency Analysis Average latency in milliseconds from when the messaging engine receives a document from the adapter until the time it is published to the message box. Reducing latency is important to some users of BizTalk, therefore tracking how much time documents spend in the inbound adapter is important.

Here is a chart showing how latency is measured.

1	2	3	4	5	6
Adapter receives message and submits	Engine receives message from adapter,	Orchestration or Solicit-Response port runs and generates a	Response message is dequeued in messaging	Messaging engine gives response message to	Adapter informs engine message is

1	2	3	4	5	6
it to the engine, work done in adapter before message is given to engine not captured in these perf counters	executes receive pipeline, map, subscription evaluation, persist message in DB.	response message.	engine, execute the send pipeline, map.	adapter.	all done.
	I				
	RR	RR	RR		
			O	O	O
				OA	OA

I = Inbound Latency

RR = Request Response Latency

O = Outbound Latency

OA = Outbound Adapter Latency

Assuming a low latency environment, this analysis checks whether the document spent more than 5 seconds in the inbound adapter. This may indicate a processing delay in the transport of messages through inbound adapters in this host instance. If multiple inbound adapters exist in this host instance, then consider separating them into their own hosts to determine which inbound adapter has high latency.

References

- BizTalk Server Database Optimization
<http://go.microsoft.com/fwlink/?linkid=104427>
- Request/Response Messaging
<http://go.microsoft.com/fwlink/?LinkID=155288>
- Identifying Bottlenecks in the BizTalk Tier
<http://go.microsoft.com/fwlink/?LinkID=155289>

BizTalk Message Delivery Throttling State Analysis

The BizTalk message delivery throttling state is one of the primary indicators of throttling. It is a flag indicating whether the system is throttling message delivery (affecting XLANG message

processing and outbound transports). The throttling condition is indicated by the numeric value of the counter. Here is a list of the values and their respective meaning:

0	Not throttling
1	Throttling due to imbalanced message delivery rate (input rate exceeds output rate)
3	Throttling due to high in-process message count
4	Throttling due to process memory pressure
5	Throttling due to system memory pressure
9	Throttling due to high thread count
10	Throttling due to user override on delivery

This analysis checks for each of these values and has a specific alert for each of them.

References

- Host Throttling Performance Counters at <http://go.microsoft.com/fwlink/?LinkID=155285>
- How BizTalk Server Implements Host Throttling at <http://go.microsoft.com/fwlink/?LinkID=155286>

BizTalk MessageBox Database Connection Failures Analysis

This performance counter is the number of attempted database connections that failed since the host instance started. If the SQL Server service hosting the BizTalk databases becomes unavailable for any reason, the database cluster transfers resources from the active computer to the passive computer. During this failover process, the BizTalk Server service instances experience database connection failures and automatically restart to reconnect to the databases. The functioning database computer (previously the passive computer) begins processing the database connections after assuming the resources during failover.

DBNetLib (Database Network Library) errors occur when the BizTalk Server runtime is unable to communicate with either the MessageBox or management databases. When this occurs, the BizTalk Server runtime instance that catches the exception shuts down and then cycles every minute to check to see whether the database is available. See the references section for more information on this topic.

When a client initiates a TCP/IP socket connection to a server, the client typically connects to a specific port on the server and requests that the server respond to the client over an ephemeral, or short lived, TCP or UDP port. On Windows Server 2003 and Windows XP, the default range of ephemeral ports used by client applications is from 1025 through 5000. Under certain conditions it is possible that the available ports in the default range will be exhausted. See the references section for more information on this topic.

This analysis checks for any occurrence of database connection failures. Database connection failures are critical because BizTalk cannot function without the database. If the cause of the database connection failure is unknown, then consider the references listed below and/or contact Microsoft Support to determine the nature of the connectivity failure.

References

- Scaled-Out Databases
<http://go.microsoft.com/fwlink/?LinkId=155307>
- Avoiding DBNETLIB Exceptions
<http://go.microsoft.com/fwlink/?LinkId=155308>
- Avoiding TCP/IP Port Exhaustion
<http://go.microsoft.com/fwlink/?LinkId=155309>

BizTalk Messaging Publishing Throttling State Analysis

The BizTalk message publishing throttling state is one of the primary indicators of throttling. It is a flag indicating whether the system is throttling message publishing (affecting XLANG message processing and inbound transports).The throttling condition is indicated by the numeric value of the counter. Here is a list of the values and their respective meaning:

0	Not throttling
2	Throttling due to imbalanced message publishing rate (input rate exceeds output rate)
4	Throttling due to process memory pressure
5	Throttling due to system memory pressure
6	Throttling due to database growth
8	Throttling due to high session count
9	Throttling due to high thread count
11	Throttling due to user override on publishing

This analysis checks for each of these values and has a specific alert for each of them.

References

- Host Throttling Performance Counters at <http://go.microsoft.com/fwlink/?LinkID=155285>
- How BizTalk Server Implements Host Throttling at <http://go.microsoft.com/fwlink/?LinkID=155286>

BizTalk Orchestrations Resident in Memory

Number of orchestration instances currently hosted by the host instance. While spikes or bursts of orchestrations resident in memory may be considered normal, an increasing trend could

indicate a “pile up” of orchestrations in memory. An increasing trend over time may occur when BizTalk is unable to dehydrate messages/orchestration instances. Try to correlate this counter with “XLANG/s Orchestrations(?)\Dehydratable orchestrations” where the question mark (?) is the same counter instance as this counter.

If a high number of orchestrations are resident in memory and if a low number of orchestrations are dehydratable, then your orchestrations are likely idle in memory and may cause a memory leak condition. Use this analysis in correlation with “XLANG/s Orchestrations(*)\Idle orchestrations” if present. An increasing trend in BizTalk Idle Orchestrations is a better indicator of memory leaks due to the inability to dehydrate orchestration instances.

This analysis checks for an increasing trend in orchestrations resident in memory and if more than 50% of the orchestrations resident in memory are not dehydratable.

References

- Orchestration Engine Performance Counters
<http://go.microsoft.com/fwlink/?LinkID=155290>
- Orchestration Dehydration and Rehydration
<http://go.microsoft.com/fwlink/?LinkID=155284>

BizTalk Private Bytes Analysis

This is the megabytes of allocated private memory for the host instance and comparable to the “\Process(*)\Private Bytes” performance counter. Private Bytes is the current size, in bytes, of memory that a process has allocated that cannot be shared with other processes. This analysis determines whether any of the host instances are consuming a large size of the system's memory and whether the host instance is increasing in memory consumption over time. A host instance consuming large portions of memory is fine as long as the it returns the memory to the system. Look for increasing trends in the chart. An increasing trend over a long period of time could indicate a memory leak.

This analysis checks for a 10 MB-per-hour increasing trend. Use this analysis in correlation with the Available Memory analysis and the Memory Leak analysis. Also, keep in mind that newly started host instances will initially appear as a memory leak when it is simply normal start up behavior. A memory leak is when a process continues to consume memory and not releasing memory over a long period of time. If you suspect a memory leak condition, then read the “Memory Growth in BizTalk Messaging” article referenced below. Otherwise, install and use the Debug Diag tool. For more information on the Debug Diag Tool, see the references section.

References

- Debug Diagnostic Tool v1.1
<http://go.microsoft.com/fwlink/?linkid=106784>
- Memory Growth in BizTalk Messaging
<http://go.microsoft.com/fwlink/?linkid=108788>

BizTalk Virtual Bytes Analysis

This is the megabytes reserved for virtual memory for the host instance. This analysis determines whether any of the host instances are consuming a large amount of the system's memory and whether the host instance is increasing in memory consumption over time. A host instance consuming large portions of memory is fine as long as it returns the memory to the system. Look for increasing trends in the chart. An increasing trend over a long period of time could indicate a memory leak.

This analysis checks for a 10 MB-per-hour increasing trend in virtual bytes. Use this analysis in correlation with the Available Memory analysis and the Memory Leak analysis. Also, keep in mind that newly started host instances will initially appear as a memory leak when it is simply normal start up behavior. A memory leak is when a process continues to consume memory and not releasing memory over a long period of time. If you suspect a memory leak condition, then read the “Memory Growth in BizTalk Messaging” article below. Otherwise, install and use the Debug Diag tool. For more information on the Debug Diag Tool, see the references section.

References

- Debug Diagnostic Tool v1.1
<http://go.microsoft.com/fwlink/?linkid=106784>
- Memory Growth in BizTalk Messaging
<http://go.microsoft.com/fwlink/?linkid=108788>

BizTalk Message Agent Database Session Throttling Analysis

This is the number of open database connections to the MessageBox compared to its respective BizTalk throttling setting. “Database connection per CPU” is the maximum number of concurrent database sessions (per CPU) allowed before throttling begins. The idle database sessions in the common per-host session pool do not add to this count, and this check is made strictly on the number of sessions actually being used by the host instance. This option is disabled by default; typically this setting should only be enabled if the database server is a bottleneck in the BizTalk Server system. You can monitor the number of active database connections by using the database session performance counter under the BizTalk:Message Agent performance object category. This parameter only affects outbound message throttling. Enter a value of 0 to disable throttling that is based on the number of database sessions. The default value is 0.

The MaxWorkerThreads registry key has influence on the number threads available to BizTalk and may help in the case where most of BizTalk’s threads are busy with database connections. This analysis checks whether the number of open database connections to the MessageBox are greater than 80 percent of the Database Session Throttling setting, indicating a throttling condition is likely.

References

- Host Throttling Performance Counters at <http://go.microsoft.com/fwlink/?LinkID=155285>
- Configuration Parameters that Affect Adapter Performance at <http://go.microsoft.com/fwlink/?LinkID=154200>

- Threads, DB sessions, and throttling at <http://go.microsoft.com/fwlink/?linkid=106793>

BizTalk Message Agent Database Session Throttling Threshold Analysis

This is the current threshold for the number of open database connections to the MessageBox. “Database connection per CPU” is the maximum number of concurrent database sessions (per CPU) allowed before throttling begins. The idle database sessions in the common per-host session pool do not add to this count, and this check is made strictly on the number of sessions actually being used by the host instance. This option is disabled by default; typically this setting should only be enabled if the database server is a bottleneck in the BizTalk Server system. You can monitor the number of active database connections by using the database session performance counter under the BizTalk:Message Agent performance object category. This parameter only affects outbound message throttling. Enter a value of 0 to disable throttling that is based on the number of database sessions. The default value is 0.

The MaxWorkerThreads registry key has influence on the number threads available to BizTalk and may help in the case where most of BizTalk’s threads are busy with database connections. This analysis checks this value to see whether it has been modified from its default setting. By default, this setting is 0, which means throttling on database sessions is disabled.

References

- Host Throttling Performance Counters at <http://go.microsoft.com/fwlink/?LinkID=155285>
- Configuration Parameters that Affect Adapter Performance at <http://go.microsoft.com/fwlink/?LinkID=154200>
- Threads, DB sessions, and throttling at <http://go.microsoft.com/fwlink/?linkid=106793>

BizTalk Message Agent In-process Message Count Throttling Analysis

This is the number of concurrent messages that the service class is processing. The “In-process messages per CPU” setting in the Host Throttling Settings is the maximum number of messages delivered to the End Point Manager (EPM) or XLANG that have not been processed. This does not include the messages retrieved from the database but still waiting for delivery in the in-memory queue. You can monitor the number of in-process messages by using the In-process message count performance counter under the BizTalk:Message Agent performance object category. This parameter provides a hint to the throttling mechanism for consideration of throttling conditions. The actual threshold is subject to self-tuning. You can verify the actual threshold by monitoring the In-process message count performance counter.

For large message scenarios (where either the average message size is high, or the processing of messages may require a large amount of messages), this parameter can be set to a smaller value. A large message scenario is indicated if memory-based throttling occurs too often and if the memory threshold gets auto-adjusted to a substantially low value. Such behavior would indicate that the outbound transport should process fewer messages concurrently to avoid excessive memory usage. Also, for scenarios where the adapter is more efficient when processing a few messages at a time (for example, when sending to a server that limits concurrent connections), this parameter may be tuned to a lower value than the default. This

analysis checks the High In-Process Message Count counter to determine whether it is greater than 80 percent of its throttling setting under the same name, which indicates a throttling condition is likely.

References

- Host Throttling Performance Counters at <http://go.microsoft.com/fwlink/?LinkID=155285>
- How to Modify the Default Host Throttling Settings at <http://go.microsoft.com/fwlink/?LinkID=155296>

BizTalk:Message Agent In-process Message Count Throttling Threshold Analysis

This is the current threshold for the number of concurrent messages that the service class is processing. The "In-process messages per CPU" setting in the Host Throttling Settings is the maximum number of messages delivered to the End Point Manager (EPM) or XLANG that have not been processed. This does not include the messages retrieved from the database but still waiting for delivery in the in-memory queue. You can monitor the number of in-process messages by using the In-process message count performance counter under the BizTalk:Message Agent performance object category. This parameter provides a hint to the throttling mechanism for consideration of throttling conditions. The actual threshold is subject to self-tuning. You can verify the actual threshold by monitoring the In-process message count performance counter.

For large message scenarios (where either the average message size is high, or the processing of messages may require a large amount of messages), this parameter can be set to a smaller value. A large message scenario is indicated if memory-based throttling occurs too often and if the memory threshold gets auto-adjusted to a substantially low value. Such behavior would indicate that the outbound transport should process fewer messages concurrently to avoid excessive memory usage. Also, for scenarios where the adapter is more efficient when processing a few messages at a time (for example, when sending to a server that limits concurrent connections), this parameter may be tuned to a lower value than the default. This analysis checks the High In-Process Message Count throttling threshold for a non-default value.

References

- Host Throttling Performance Counters at <http://go.microsoft.com/fwlink/?LinkID=155285>
- How to Modify the Default Host Throttling Settings at <http://go.microsoft.com/fwlink/?LinkID=155296>

BizTalk Message Agent Process Memory Usage (MB) Throttling Analysis

This is the memory usage of current process (MB). BizTalk process memory throttling can occur if the batch to be published has steep memory requirements, or if too many threads are processing messages. If the system appears to be over-throttling, consider increasing the value associated with the process memory usage threshold for the host and verify that the host instance does not generate an "out of memory" error. If an "out of memory" error is raised by increasing the process memory usage threshold, then consider reducing the values for the internal message queue size

and in-process messages per CPU thresholds. This strategy is particularly relevant in large message processing scenarios.

If your BizTalk server regularly runs out of virtual memory, then consider BizTalk Server 64-bit. Each Process on 64-bit servers can address up to 4 TB's of virtual memory versus the 2 GB's in 32-bit. In general, 64-bit BizTalk and 64-bit SQL Server is highly recommended. See the "BizTalk Server 64-bit Support" reference for more information. This analysis checks whether the process memory usage is greater than 80 percent of its respective throttling threshold of the same name. By default, the BizTalk Process Memory Usage throttling setting is 25 percent of the virtual memory available to the process. The /3GB switch has no effect on this setting.

References

- How BizTalk Server Implements Host Throttling at <http://go.microsoft.com/fwlink/?LinkID=155286>
- How to Modify the Default Host Throttling Settings at <http://go.microsoft.com/fwlink/?LinkID=155296>
- How to Capture a Memory Dump of a Process that is Leaking Memory at <http://go.microsoft.com/fwlink/?LinkId=155305>
- BizTalk Server 64-bit Support at <http://go.microsoft.com/fwlink/?LinkId=155306>

BizTalk:Message Agent Process Memory Usage (MB) Throttling Threshold Analysis

This is the current threshold for the memory usage of current process (MB). The threshold may be dynamically adjusted depending on the actual amount of memory available to this process and its memory consumption pattern. BizTalk process memory throttling can occur if the batch to be published has steep memory requirements, or if too many threads are processing messages. If the system appears to be over-throttling, consider increasing the value associated with the process memory usage threshold for the host and verify that the host instance does not generate an "out of memory" error. If an "out of memory" error is raised by increasing the process memory usage threshold, then consider reducing the values for the internal message queue size and in-process messages per CPU thresholds. This strategy is particularly relevant in large message processing scenarios.

If your BizTalk server regularly runs out of virtual memory, then consider BizTalk Server 64-bit. Each Process on 64-bit servers can address up to 4 TB's of virtual memory versus the 2 GB's in 32-bit. In general, 64-bit BizTalk and 64-bit SQL Server is highly recommended. See the "BizTalk Server 64-bit Support" reference for more information. This analysis checks whether the Process memory throttling is set to a non-default value.

References

- How BizTalk Server Implements Host Throttling at <http://go.microsoft.com/fwlink/?LinkID=155286>
- How to Modify the Default Host Throttling Settings at <http://go.microsoft.com/fwlink/?LinkID=155296>

- How to Capture a Memory Dump of a Process that is Leaking Memory at <http://go.microsoft.com/fwlink/?LinkId=155305>
- BizTalk Server 64-bit Support at <http://go.microsoft.com/fwlink/?LinkId=155306>

Monitoring BizTalk Server with System Center Operations Manager 2007

Monitoring your BizTalk applications and infrastructure with Microsoft System Center Operations Manager (Operations Manager) is the preferred monitoring approach. The Microsoft BizTalk Server 2010 management packs for Operations Manager provide proactive and reactive monitoring of computers running BizTalk Server. These management packs provide dozens of built-in, customizable rules to allow for comprehensive and automated monitoring of BizTalk Server.

The following BizTalk Server management pack is available for use with Operations Manager:

- [BizTalk Server 2010 Monitoring Management Pack](http://go.microsoft.com/fwlink/?LinkId=210666)
(<http://go.microsoft.com/fwlink/?LinkId=210666>).

This section of the operations guide includes background information, best practices, checklists, and procedures that you can use to assist in implementing a monitoring strategy based on Operations Manager.

In This Section

- [Best Practices for Monitoring with Operations Manager 2007](#)
- [Monitoring Throttling Using Performance Threshold Rules](#)
- [Monitoring SQL Servers](#)
- [Monitoring Applications and Host Instances](#)

Best Practices for Monitoring with Operations Manager 2007

Adhere to the best practices listed in this topic to effectively monitor your applications using Operations Manager 2007.

Install additional management packs for more complete coverage

- To help ensure that Operations Manager 2007 monitors the key applications in your BizTalk Server environment, you should install the following management packs:
 - BizTalk Server (required)
 - Windows Server Operating System (optional)
 - Microsoft Windows Server Failover Cluster (if clusters are used, optional)
 - SQL Server Monitoring
 - Internet Information Services 7
 - Message Queuing (MSMQ) 5.0 (optional)

Review and prioritize alerts on a daily basis

- Reviewing and prioritizing alerts on a daily basis helps to ensure that issues are resolved in a timely manner.

Verify that BizTalk Server is communicating with the Operations Manager 2007 server.

- If communication fails between the servers running BizTalk Server and the monitoring infrastructure, you will not receive alerts.

Enable and disable rules as necessary

- By default, some of the rules in the BizTalk Server management pack are disabled. These disabled rules are of the following types: rules needing customization, rules that serve as templates, and rules for monitoring additional BizTalk Server events. Make sure the relevant rules for your BizTalk Server environment are enabled.

Customize rules as necessary for your environment

- You should customize the rules in the BizTalk Server management pack to suit your BizTalk Server deployment. Some rules require monitoring thresholds or time thresholds that are best defined based on your specific BizTalk Server deployment.

Create additional rules as necessary, based on the rules included in the BizTalk Server Management Pack

- Rules are provided for use as templates for artifacts that you create, such as BizTalk Server hosts. You should use these template rules as a reference when creating artifact-specific rules such as:
 - Host-specific rules, for example, host queue monitoring, and host throttling monitoring
 - MessageBox-specific rules
 - Rules for additional third party components, for example, MQSeries adapter

Monitor all the BizTalk Server related components

The components of the BizTalk Server environment that you should monitor to ensure that they are running include, but are not necessarily limited to:

- BizTalk Host instances
- SQL Server Agent jobs installed with BizTalk Server
- Any custom services developed for the BizTalk solution
- Status of any custom databases developed for the BizTalk solution
- Any custom event log entries relevant to the BizTalk Server environment

See Also

[Monitoring BizTalk Server with System Center Operations Manager 2007](#)

Monitoring Throttling Using Performance Threshold Rules

BizTalk Server will initialize throttling to protect the system from reaching an unrecoverable state. Throttling can indicate a problem and assist you in identifying its source. After you have identified the cause of the bottleneck based on the throttling state, analyze the other performance counters to narrow down the source of the problem.

For example, high contention on the MessageBox database could be due to high CPU usage, which could be caused by excessively paging to disk, which in turn could be caused by low memory conditions. High contention on the MessageBox could also be caused by high lock contention, which could be due to saturated disk drives.

Monitoring the Message Delivery Throttling State and the Message Publishing Throttling State for each host instance is usually a good place to start when troubleshooting throttling. If the value of these counters is not zero, it is indicative that throttling is happening within the BizTalk Server system and it is possible to further analyze the cause of the bottleneck. For descriptions on the other performance counters, see [Identifying Bottlenecks in the Database Tier](http://go.microsoft.com/fwlink/?LinkID=154678) (<http://go.microsoft.com/fwlink/?LinkID=154678>) in BizTalk Server Help.

BizTalk Server System Performance Counters

Object	Instance	Counter	Monitoring Purpose
Processor	_Total	% Processor Time	Resource Contention
Process	BTSNTSvc	Virtual Bytes	Memory Leak/Bloat
Process	BTSNTSvc	Private Bytes	Memory Leak/Bloat
Process	BTSNTSvc	Handle Count	Resource Contention
Process	BTSNTSvc	Thread Count	Resource Contention
Physical Disk	_Total	% Idle Time	Resource Contention
Physical Disk	_Total	Current Disk Queue Length	Resource Contention

BizTalk Application Counters

Object	Instance	Counter	Description
BizTalk Messaging	RxHost	Documents Received/Sec	Incoming Rate
BizTalk Messaging	TxHost	Documents Processed/Sec	Outgoing Rate
XLANGs/Orchestrations	PxHost	Orchestrations Completed/Sec	Processing Rate
BizTalk: MessageBox: General Counters	MsgBoxName	Spool Size	Cumulative size of all Host Queues
BizTalk: MessageBox:	MsgBoxName	Tracking Data Size	Size of

Object	Instance	Counter	Description
General Counters			TrackingData table on the MessageBox
BizTalk: MessageBox: Host Counters	PxHost:MsgBoxName	Host Queue - Length	Number of messages in the specific Host Queue
BizTalk: MessageBox: Host Counters	TxHost:MsgBoxName	Host Queue - Length	Number of messages in the specific Host Queue
BizTalk: Message Agent	RxHost	Database Size	Size of publishing (PxHost) Queue
BizTalk: Message Agent	PxHost	Database Size	Size of publishing (TxHost) Queue
BizTalk: Message Agent	HostName	Message Delivery Throttling State	Affects XLANG and Outbound transports
BizTalk: Message Agent	HostName	Message Publishing Throttling State	Affects XLANG and Inbound transports

In This Section

- [Monitoring for Throttling](#)

Monitoring for Throttling

The BizTalk Server management pack monitors performance counters that indicate the throttling state of BizTalk Server. Some key factors to understand about throttling are listed below.

- Rate-based throttling is per host and is based on the rate of messages inbound vs. outbound messages.
- For delivery throttling (**MsgBox -> Send Port or Orchestration**), inbound rate is the rate at which messages are received from the message box. Outbound rate is the rate at which messages are successfully delivered via the adapters.
- For publishing throttling (**Receive adapters or Orchestrations -> MsgBox**), inbound rate is the rate at which messages are received from the adapters and outbound rate is the rate messages are plugged into the MsgBox.
- No throttling mechanism exists between hosts other than total messages in the database.

For additional background information, refer to the topic [How BizTalk Server Implements Host Throttling](http://go.microsoft.com/fwlink/?LinkID=155286) (http://go.microsoft.com/fwlink/?LinkID=155286) in BizTalk Server Help.

BizTalk Server incorporates self-throttling, which helps to prevent overloading of the server based on various parameters. A temporary overload that causes throttling to occur is not an operationally significant event. Persistent throttling, however, is not expected in a stable environment and could indicate underlying problems at the infrastructure level. The management pack provides proactive monitoring of such persistent throttling conditions with performance threshold rules.

Four utilization/performance tracking rules monitor for extended periods of throttling caused by four different conditions as indicated in the following table.

Condition	Rule
BizTalk Server service process memory	Warning: BizTalk Throttled on High Process Memory for a significant period
Number of messages being processed	Warning: BizTalk Throttled on High Inprocess Message Count for a significant period
Number of threads in a BizTalk Server process	Warning: BizTalk Throttled on High Thread Count for a significant period
Size of the BizTalk Server database queues	Warning: BizTalk Throttled on High Database Size for a significant period

These threshold rules use data providers based on throttling state indicator performance counters. For more information about these performance counters, refer to the section [Performance Counters](http://go.microsoft.com/fwlink/?LinkID=157269) (http://go.microsoft.com/fwlink/?LinkID=157269) in BizTalk Server Help.

These rules are configured to raise an alert if the average of over a certain number of samples crosses a particular threshold (default is 30). For example, “Warning: BizTalk Throttled on High Database Size for a significant period” is a rule monitoring the throttling state of all BizTalk Server processes in a given computer. This rule uses a data provider based on the throttling state indicator performance counter “BizTalk:Message Agent-High database size.” If this performance counter value is 1, then the associated process is throttling because of high database size.

The preceding rule is configured to take an average of 30 samples and raise an alert if the average of the samples is more than 0.6. Since each sample is taken at an interval of one minute, this implies that over the past 30 minutes, at least one or more BizTalk Server processes in that computer were throttling because of high database size, 60 percent of the time.

This heuristic may not suit your particular application scenario. Based on the historical behavior in your environment as described before, you should configure these rules with the correct values by either:

- Adjusting the samples.
- Adjusting the threshold value.

- If necessary, modifying the interval of sampling for the provider.

Monitoring SQL Servers

The Microsoft SQL Server management pack provides both proactive and reactive monitoring of SQL Server 2008 R2 and SQL Server 2008 SP1 in an enterprise environment. Availability and configuration monitoring, performance data collection, and default thresholds are built for enterprise-level monitoring. Both local and remote connectivity checks help ensure database availability.

With the embedded expertise in the SQL Server management pack, you can proactively manage SQL Server, and identify issues before they become critical. This management pack increases the security, availability, and performance of your SQL Server infrastructure.

The Microsoft SQL Server management pack guide that comes with the pack describes the content of the management pack, and describes how to deploy it. Features of the management pack include:

- Monitoring the state of the included services such as SQL Server, SQL Agent, Report Server, Notification Services
- Monitoring the state of databases
- Monitoring the available space in databases, configurable by % or MB
- Ensuring databases are configured correctly
- Ensure clients can connect to the SQL Server
- Monitor blocked processes
- Watch for failed agent jobs, and jobs taking an excessive time to execute
- Monitor the health of replication and alert on failures
- Monitor the state of Database Mirroring

In This Section

- [Monitoring SQL Server Agent Jobs and Databases](#)
- [How to Mark BizTalk Server Databases for Customized Monitoring](#)
- [Monitor the BizTalk Server Databases](#)

Monitoring SQL Server Agent Jobs and Databases

BizTalk Server includes multiple SQL Server Agent jobs that perform important functions to keep your servers operational and healthy. You should monitor the health of these jobs and ensure that they are running without errors. The Microsoft SQL Server Management Pack contains rules for monitoring items such as SQL databases and SQL Server Agent jobs. You should configure the SQL Server Management Pack for comprehensive monitoring of all BizTalk Server databases and SQL Server Agent jobs.

The BizTalk Server management pack includes two disabled rules for monitoring the health of two of the most important BizTalk SQL Server Agent jobs:

- Critical Error: A BizTalk SQL Server Agent job failed - Backup BizTalk Server
- Critical Error: A BizTalk SQL Server Agent job failed – Tracked Message Copy

To monitor all BizTalk Server SQL Server Agent jobs from within the BizTalk Server management pack, you must enable these rules and create additional rules for other jobs that you want to monitor using the following process.

- In the Operations Manager Administrator console, create a copy of either of the two preceding rules in the BizTalk Server Core Rule group, and rename the rule appropriately.
- In the criteria section for the rule, change the wildcard comparison for Parameter 1 appropriately.
- In some cases, job names are dependent on database names that the user creates, for example, the name of the MessageBox database.
- Your rule can be targeted either towards a job associated with a single MessageBox or all MessageBoxes.

In This Section

- [How to Start the SQL Server Agent](#)

How to Start the SQL Server Agent

For information about how to start the SQL Server agent, see [How to: Start SQL Server Agent \(SQL Server Configuration Manager\)](#) (<http://go.microsoft.com/fwlink/?LinkId=154672>) in SQL Server Books Online.

How to Mark BizTalk Server Databases for Customized Monitoring

If you have installed the Microsoft SQL Server Management Pack, you can customize the way BizTalk Server databases are monitored. This ensures that the SQL Server Management Pack monitors the following BizTalk Server databases:

- BAM Archive (BAMArchive) database
- BAM Primary Import (BAMPrimaryImport) database
- BAM Star Schema (BAMStarSchema) database
- BizTalk Tracking (BizTalkDTADb) database
- BizTalk Management (BizTalkMgmtDb) database
- BizTalk MessageBox (BizTalkMsgBoxDb) database
- Rule Engine (BizTalkRuleEngineDb) database
- Enterprise Single Sign-On (SSODB) database



Note

You must be logged on as a member of the Operations Manager Advanced Operator role or Operations Manager 2007 Management Group.

► To mark BizTalk Server databases for customized monitoring

1. Click **Start**, click **All Programs**, click **System Center Operations Manager 2007**, and then click **Operations Console**.
2. In the Operations console, click the **Authoring** button.
3. In the **Authoring** pane, expand **Management Pack Objects**, and then click **Object Discoveries**.

4. To locate a discovery for SQL Server, on the Operations console toolbar click **Find**, and in the **Look for** text box enter **SQL 2008** to search for SQL Server 2008.
5. Under the **Discovered Type: SQL 2008 DB** category, select **Discover Databases for a Database Engine**.
6. On the Operations console toolbar, click **Overrides** and then point to **Override the Object Discovery**. You can choose to override the discovery for objects of a specific type or for all objects within a group. After you choose which group of object type to override, the **Override Properties** dialog box opens.



Note

If the **Overrides** button is not available, make sure you have selected a monitor and not a container object in the Monitors pane.

7. Select the check box in the **Override** column next to the **Exclude List** parameter, and in the **Override Value** column, type the name of the database that you want to exclude from monitoring. Make sure the BizTalk Server databases that you want to monitor are not listed in the **Override Value** column.



Tip

You can use the modified object discoveries to create a new management pack. At the bottom of the pane, the **Select destination management pack** drop-down shows a default value of **Default Management Pack**. Click **New** to create a new management pack using the modified object discoveries.

See Also

[Monitoring SQL Servers](#)

Monitor the BizTalk Server Databases

You can run the Monitor BizTalk Server SQL Agent job to identify any known issues in Management, Message Box, or DTA databases. The job is created when you configure a BizTalk group in BizTalk Server Administration console or upgrade BizTalk from the previous version.

The Monitor BizTalk Server Job

The Monitor BizTalk Server job scans for the following issues in Management, Message Box, and DTA databases:



Note

The Monitor BizTalk Server job only scans for issues. It does not fix the issues found.

- Messages without any references
- Messages without reference counts
- Messages with reference count less than 0
- Message references without spool rows
- Message references without instances
- Instance state without instances
- Instance subscriptions without corresponding instances

- Orphaned DTA service instances
- Orphaned DTA service instance exceptions
- TDDS is not running on any host instance when global tracking option is enabled

The Monitor BizTalk Server job is configured and automated to run once in a week. Since the job is computationally intensive, we recommended that you schedule it to run during periods of downtime or low traffic.

The job fails if it encounters any issues; the error string returned contains the number of issues found. Else, it runs successfully. You can see the details in the job history. If you run the job with Administrator privileges, the error string will be logged to both the job history and the SQL Server Application log.

Important

Failure of this job does not necessarily constitute a critical issue, but rather an issue that should be investigated and addressed as part of regular maintenance of the BizTalk Server databases. This job fails by design in the event it discovers one of the issues listed above. For more information about the issues listed above and to access utilities that are commonly used by Microsoft Product Support Services to troubleshoot these issues please see [Using BizTalk Terminator to resolve issues identified by BizTalk MsgBoxViewer](http://go.microsoft.com/fwlink/?LinkId=210367) (<http://go.microsoft.com/fwlink/?LinkId=210367>).

Monitoring Applications and Host Instances

This section describes monitoring BizTalk applications and host instances using Operations Manager 2010. Running the Performance Analysis of Logs (PAL) tool also can provide various details that can assist you in monitoring your applications and host instances. For more information on this tool, see [Using the Performance Analysis of Logs \(PAL\) Tool](#).

In This Section

- [Monitoring Applications](#)
- [Monitoring Host Instances](#)

See Also

[Using the Performance Analysis of Logs \(PAL\) Tool](#)

Monitoring Applications

Setting up Microsoft System Center Operations Manager to monitor BizTalk applications typically can be broken down into a progressive four-step process as follows:

1. **Modify existing rules and/or copy rules to a custom management pack to monitor a custom BizTalk application**

You will need to copy many of these rules multiple times. This is the case if you are monitoring a number of file shares, for example. In this scenario, you would copy the base rule once for each file share with the file share address added in the description field on the **Criteria** tab of the rule. By adding the address, Operations Manager will raise an alert for

each individual file share. When you copy an existing rule, ensure that you select **Enable** rule-disable overrides for this rule or you will receive duplicate alerts.

Another item that you should add to each rule that you create is Knowledge information on the **Knowledge Base** tab of the rule property. This data is attached to the notification that is raised when Operations Manager sends out an alert. The power of this feature becomes clear when you include steps that may help resolve the error.

2. **Create actions for each defined rule**

Creating or copying a rule is really the first step in the process. The next step is to take some action based on that rule. If there is no action based on a rule then it doesn't really matter that the event was ever monitored. The action most frequently taken is to alert an operator or administrator that an error has occurred. Operations Manager also provides a number of other actions that can be used when an event is triggered. These actions include:

- Starting a script
- Sending a Simple Network Management Protocol (SNMP) trap (in SNMP, agents monitor the activity in the various devices on the network and report to the network console workstation)
- Sending a notification to a Notification group
- Executing a command or batch file
- Updating a state variable
- Transferring a file
- Calling a method on a managed code assembly

3. **Create iterative processes to automate manual tasks**

The next step is an iterative process and moves beyond the basic alerting mechanism. With Operations Manager able to invoke both script and .NET code, the process of automating manual tasks based on raised events is a powerful and time saving feature. An example of this is to run a script to automatically start a port if a disabled /stopped event message is logged. This process is iterative since many processes can be automated.

4. **Use threshold rules to automate manual tasks**

The next step in processing is to move beyond the reactive alerting and use threshold rules. The BizTalk Server 2010 Management Pack does not contain any threshold rules by default. This is because such rules are typically specific to the custom application and are different for every application. A threshold embodies a business rule regarding the custom application and provides a proactive means of monitoring a system. You can use the BizTalk Server threshold templates provided with the Performance Analysis of Logs (PAL) tool to define rules.

An example of such a threshold rule is to measure when the CPUs on a server consistently run above 75 percent for a specific time period. This could indicate that you need to scale out the system. Yet another example is where you create a threshold rule that monitors a unique set of counters. This rule could then invoke code to initialize BizTalk host instances on a previously configured backup server during periods of high demand.

See Also

Monitoring Host Instances

This topic describes monitoring BizTalk host instances using Microsoft System Center Operations Manager.

Using Threshold Rules to Monitor Health

The BizTalk Server 2010 Management Pack incorporates performance threshold rules that provide a comprehensive view of the health of the BizTalk hosts. Two different types of threshold rules are provided:

- Rules that apply generically (for example, to all BizTalk hosts and to all MessageBox databases).
- Rules that are specific to a particular BizTalk host.

Generic rules monitor all the BizTalk hosts based on a common threshold. For example, the rule Monitor HostQ Size monitors the work queues of all BizTalk hosts based on a common threshold. If there are three different hosts, all their work queues are monitored by the same rule, and alerts occur when any of the host work queues cross the common threshold.

BizTalk host-specific rules enable you to configure different thresholds for different hosts. For example, the rule Monitor HostQ Size – BizTalkServerApplication is a host-specific rule that monitors the work queue of the BizTalkServerApplication host. In this example you can define a specific Operations Manager provider for the particular performance counter instance and use that provider in the threshold rule. Because these rules are host-specific, you must define rules specific to each newly created host.

BizTalk host-specific rules are provided as template rules for creating rules that are applicable in your environment. All threshold monitoring rules are disabled by default:

- You should configure generic rules with threshold values specific to your environment.
- You should create BizTalk host-specific rules based on the template rules and appropriate thresholds.

Monitoring BizTalk Host Instances

Rules that target specific BizTalk hosts are more flexible from a monitoring perspective. All threshold monitoring rules for the BizTalkServerApplication host provided in the BizTalk Server 2010 Management pack are template rules. In order to use these rules, you should use the Operations Manager Administrator console to:

- Create a copy of the template rule in the BizTalk Server rule group and rename it.
- Create a new Operations Manager provider for the BizTalk host-specific performance counter instance.
- Modify the Operations Manager provider used by the rule and point it to the new one.

Suppose your BizTalk Server installation has a BizTalk host ReceiveHost and you want to monitor the Host Queue size for this host. In this case, you should create a Operations Manager provider based on the ReceiveHost instance of the performance counter for the queue size for the BizTalk host. You should also set the threshold value in the rule appropriately for your environment.

If you are using host-specific threshold monitoring rules, you should disable generic monitoring rules. This prevents redundant alerts.

See Also

[Monitoring BizTalk Server with System Center Operations Manager 2007](#)

Maintaining BizTalk Server

Maintenance activities are part of the Service Monitoring and Control (SMC) system management function as defined by the Microsoft Operations Framework (MOF). The primary goal of SMC is to observe the health of your BizTalk Server system. SMC checks may initiate remedial actions to avoid potential service incidents and to minimize the impact of service incidents when they do occur.

For the purposes of this section of the guide, SMC activities are divided into four three broad categories: reliability, administration, security, and performance. The daily, weekly, and monthly maintenance checklists organize SMC checks according to how frequently each check should be performed. If SMC checks indicate that remedial action is necessary, the checklists will direct you to sources of additional information to resolve the problem.

The sections pertaining to resolving reliability, administration, security, and performance issues contain information about identifying and resolving many common problems encountered during BizTalk Server operations.

For more information about the Microsoft Operations Framework, see <http://go.microsoft.com/fwlink/?linkid=88047>. For more information about the Service Monitoring and Control function, see <http://go.microsoft.com/fwlink/?LinkId=155341>.

In This Section

- [Maintaining Reliability](#)
- [Administrative Maintenance](#)
- [Maintaining Performance](#)
- [Maintaining BizTalk Server Databases](#)

Related Sections

[Checklist: Performing Daily Maintenance Checks](#)

[Checklist: Performing Weekly Maintenance Checks](#)

[Checklist: Performing Monthly Maintenance Checks](#)

Maintaining Reliability

This section provides information about how you can resolve reliability issues with a BizTalk Server system. These issues may be discovered by the routine maintenance checks that are performed in the [Routine Maintenance Checklists](#) section of this document.

In addition to the topics in this section, other topics in this document address reliability issues. These topics are listed in [Related Sections](#) below.

Testing Group Failover

Perform the procedures in this section as part of the reliability checks that should be performed monthly. These procedures include testing the group failover policy, and testing whether the group resources can fail over.



Note

If the computer is joined to a domain, members of the Domain Admins group should be able to perform this procedure.



Note

To perform the procedures in this section, you must be logged on as a member of the Administrators group on the local computer, or you must have been delegated the appropriate authority.

Perform the following steps to test group failover on computers running Windows Server 2008.

► To test a group failover policy

1. Make sure you have installed the **Failover Clustering** feature on at least two computer running Windows Server 2008 so as to create a two node Windows Failover Cluster. For instructions on how to install this feature, see [Install the Failover Clustering Feature](http://go.microsoft.com/fwlink/?LinkId=157259) (<http://go.microsoft.com/fwlink/?LinkId=157259>).
2. Open Failover Cluster Management by clicking **Start**, clicking **Administrative Tools**, and then clicking **Failover Cluster Management**.
3. In the console tree, expand the cluster node, expand the **Services and Applications** node, right-click the clustered instance of the application to be failed over, and then click **Properties**.
4. On the **Failover** tab, set **Maximum failures in the specified period** to 0, and then click **OK**.
5. In the console tree, expand the **Services and Applications** node.
6. In the details pane, right-click a resource, and then click **Properties**.
7. On the **Policies** tab, set **Maximum restarts in the specified period** to 0, and then click **OK**.
8. Right-click the resource, click **More Actions**, and then click **Simulate Failure of this resource**. Verify whether the group reacts based on the policy you specified in the previous step.
9. Right-click the clustered instance of the application to be failed over, and then click **Properties**.
10. On the **Failover** tab, set **Maximum failures in the specified period** to 1, and then click **OK**.
11. Right-click the resource and select **Bring this resource online**.

► **To test whether group resources can fail over**

1. Make sure you have installed the **Failover Clustering** feature on the computer running Windows Server 2008. For instructions on how to install this feature, see [Install the Failover Clustering Feature](http://go.microsoft.com/fwlink/?LinkId=157259) (<http://go.microsoft.com/fwlink/?LinkId=157259>).
2. Open Failover Cluster Management by clicking **Start**, clicking **Administrative Tools**, and then clicking **Failover Cluster Management**.
3. In the console tree, expand the cluster node, expand the **Services and Applications** node, and then click the clustered instance of the application to be failed over.
4. On the **Action** menu, click **Move this service or application to another node**, and then click the node to which the application will be failed over.
5. In the **Please confirm action** dialog box, choose to move the application to selected node.
6. Make sure that the node to which you moved the application to is listed against the **Current Owner** in the details pane of the application.

Ensuring Multiple Servers Are Part of a BizTalk Group

To ensure the reliability of a system, at least two physical BizTalk servers should be part of the BizTalk group. If you need to add a server to a BizTalk group, keep the following in mind:

- A server can only be associated with one BizTalk group. If a server already belongs to another group, you must first remove that server from its current group before you can add it to a new group. For more information about removing a server from a BizTalk group, see "How to Remove a Server from a Group" in BizTalk Server Help at <http://go.microsoft.com/fwlink/?LinkId=155577>.
- BizTalk groups associated with different servers in a BizTalk Server environment do not interact except to exchange messages.
- The BizTalk Server runtime must be installed on the computer you want to add to the BizTalk group.



Note

To perform the procedures in this topic, you must be logged on as a member of the SSO Administrators group and as a member of the Windows Administrators group.

► **To determine whether at least two physical BizTalk servers are part of the BizTalk group**

1. Open the BizTalk Server Administration console by clicking **Start**, pointing to **All Programs**, pointing to **Microsoft BizTalk Server 2010**, and then clicking **BizTalk Server Administration**.
2. Expand the **BizTalk Server 2010** node, expand the **BizTalk Group** node, and then expand the **Platform Settings** node.
3. Click the **Servers** node. Verify that more than one server is listed in the **Servers** pane.



Note

To view information about the server, right-click the server, point to **View**, and then click **Group Hub Page**.

► To add a server to a BizTalk group

1. On the computer that you want to add to a BizTalk group, click **Start**, click **All Programs**, click **Microsoft BizTalk Server 2010**, and then click **BizTalk Server Configuration**.
2. In **Microsoft BizTalk Server 2010 Configuration**, select **Custom configuration**.
3. In **Database server name**, type the name of the SQL server for the BizTalk group that the server is joining.
4. In **Service credential**, type the appropriate user name and password that the services will use, and then click **Configure**.
5. In the navigation tree on the left side of the screen, click **Enterprise SSO**.
6. On the **Enterprise Single Sign-On** page, click **Join an existing SSO system**.



Note

Ensure that the server name and database name point to the master SSO database server for the BizTalk group that the server is joining.

7. In the navigation tree on the left side of the screen, click **Group**.
8. On the **Group** page, click **Join an existing BizTalk Group**.



Note

Ensure that the server name and database name point to the databases for the BizTalk group that the server is joining.

9. On the menu bar, click **Apply Configuration** to configure both Enterprise Single Sign-On and the group on this computer.

Related Sections

- For information about checking for failed disks in the hardware RAID, see "View Disk Properties" in the Windows Server 2003 product Help at <http://go.microsoft.com/fwlink/?linkid=104161>.
- For information about manually checking for suspended messages, see "Investigating Orchestration, Port, and Message Failures" in BizTalk Server Help at <http://go.microsoft.com/fwlink/?LinkID=154512>.
- For information about ensuring that each host has an instance running on at least two physical BizTalk servers, see [High Availability for BizTalk Hosts](#).
- For information about ensuring that each host has an instance running on at least two physical BizTalk servers, see [Scaling Out Receiving Hosts](#).
- For information about ensuring that failover for all clustered services has been tested, see [Clustering the Master Secret Server](#).
- For information about ensuring that the BizTalk databases are clustered under SQL Services, see [Clustering the BizTalk Server Databases](#).

- For information about determining whether any unstable code is being used (requiring separate hosts), see [High Availability for BizTalk Hosts](#).
- For information about performing functional testing of all new BizTalk applications, see [Testing an Application](#)

Administrative Maintenance

This section provides information about how you can resolve administration issues with a BizTalk Server system. These issues may be discovered by the routine maintenance checks that are performed in the [Routine Maintenance Checklists](#) section.

In addition to the topics in this section, other topics in this document address administration issues. These topics are listed in Related Sections below.

In This Section

- [Best Practices for Administrative Maintenance](#)
- [How to Start the SQL Server Agent](#)
- [How to Schedule a Backup BizTalk Server Job](#)
- [How to Configure a Backup BizTalk Server Job](#)

Related Sections

- For information about ensuring that the SQL Server Agent service is running on the SQL Server, and ensuring that all BizTalk related SQL Server jobs are working properly, see [Monitoring SQL Server Agent Jobs](#).
- For information about removing a BizTalk application or artifact using the BTSTask command-line tool, see "RemoveApp Command" in BizTalk Server Help at <http://go.microsoft.com/fwlink/?LinkID=154687>.
- For information about removing an artifact from an application using either the BizTalk Server Administration console or the BTSTask command-line tool, see "How to Remove an Artifact from an Application" in BizTalk Server Help at <http://go.microsoft.com/fwlink/?LinkID=154688>.
- For information about verifying the configuration in the BizTalk Administration Console, see "Using the BizTalk Server Administration Console" in BizTalk Server Help at <http://go.microsoft.com/fwlink/?LinkID=154689>.
- For information about verifying the BTSNTSvc.exe.config file, see "BTSNTSvc.exe.config File" in BizTalk Server Help at <http://go.microsoft.com/fwlink/?LinkID=154690>.
- For information about verifying the BizTalk related registry keys, see "Windows registry information for advanced users" in the Microsoft Help and Support Web site at <http://go.microsoft.com/fwlink/?LinkId=155583>.
- For information about running the BizTalk Best Practices Analyzer, see "BizTalk Server Best Practices Analyzer" in the Microsoft Download Center at <http://go.microsoft.com/fwlink/?LinkId=83317>.
- For information about maintaining BAM databases, see "Managing BAM Databases" in BizTalk Server Help at <http://go.microsoft.com/fwlink/?LinkId=155584>.

- For information about ensuring that the latest service packs and hotfixes are installed, see the Microsoft Update Web site at <http://go.microsoft.com/fwlink/?LinkID=47813>.

Best Practices for Administrative Maintenance

This topic points you to the best practices (in the form of checklists) that you should follow when resolving administration issues with a BizTalk Server system.

For daily maintenance checks	Checklist: Performing Daily Maintenance Checks
For weekly maintenance checks	Checklist: Performing Weekly Maintenance Checks
For monthly maintenance checks	Checklist: Performing Monthly Maintenance Checks

How to Start the SQL Server Agent

For information about how to start the SQL Server agent, see [How to: Start SQL Server Agent \(SQL Server Configuration Manager\)](#) (<http://go.microsoft.com/fwlink/?LinkId=154672>) in SQL Server Books Online.

How to Schedule a Backup BizTalk Server Job

The Backup BizTalk Server job runs as scheduled by the SQL Server Agent service. If you want to create more frequent or less frequent backups, you can change the schedule of the Backup BizTalk Server job by using SQL Server Enterprise Manager. For more information about how to schedule a Backup BizTalk Server job, see [How to Schedule the Backup BizTalk Server Job](#) (<http://go.microsoft.com/fwlink/?LinkId=154674>) in BizTalk Server Help.

How to Configure a Backup BizTalk Server Job

You must configure the Backup BizTalk Server job before you can back up BizTalk Server. For more information about how to configure a Backup BizTalk Server job, see [How to Configure the Backup BizTalk Server Job](#) (<http://go.microsoft.com/fwlink/?LinkID=153813>) in BizTalk Server Help.

Maintaining Performance

This section provides information that is intended to help you resolve performance issues discovered during your routine maintenance checks. You can also use the tools and techniques described here proactively, to identify potential problems before they become critical issues.

You will generally need to establish a performance baseline as a standard against which to assess current system performance.

In addition to the topics in this section, other topics in this document address performance issues. These topics are listed in Related Sections below.

In This Section

- [Best Practices for Maintaining Performance](#)
- [Configuring Batching to Improve Adapter Performance](#)
- [How to Adjust the Message Size Threshold](#)
- [How to Adjust the Configuration Cache Refresh Interval](#)
- [How to Disable Tracking](#)
- [Troubleshooting Performance Issues](#)

Related Sections

- For more information about performance issues in general, see BizTalk Server Performance Optimization Guide at <http://go.microsoft.com/fwlink/?LinkID=150492>.
- For information about analyzing weekly performance monitor logs against baseline and thresholds, see [Using the Performance Analysis of Logs \(PAL\) Tool](#), "Finding and Eliminating Bottlenecks" in the BizTalk Server Performance Optimization Guide at <http://go.microsoft.com/fwlink/?LinkId=154675>, and [Troubleshooting Performance Issues](#).
- For information about ensuring that the system is not experiencing frequent auto-growth of BizTalk Server databases, see [Defining Auto-Growth Settings for Databases](#), "Tracking Database Sizing Guidelines" in the BizTalk Server Help at <http://go.microsoft.com/fwlink/?LinkId=154677>, and "Identifying Bottlenecks in the Database Tier" in the BizTalk Server Help at <http://go.microsoft.com/fwlink/?LinkId=154678>.
- For information about maintaining SQL Server, [Performing SQL Server Maintenance Procedures](#).
- For information about running SQL Server Profiler during high load to check for long response times and high resource usage, see "Using SQL Server Profiler" in SQL Server Help at <http://go.microsoft.com/fwlink/?LinkID=106720>.
- For information about ensuring that message batching for all adapters is appropriate for resource consumption or latency, see [Configuring Batching to Improve Adapter Performance](#).
- For information about ensuring that the large message threshold is appropriate for resource consumption, see [How to Adjust the Message Size Threshold](#).

- For information about increasing the BizTalk Server cache refresh interval, see [How to Adjust the Configuration Cache Refresh Interval](#).
- For information about inbound and outbound host throttling, see "What is Host Throttling?" in BizTalk Server Help at <http://go.microsoft.com/fwlink/?LinkId=154694>. For information about triggers, actions, and mitigation strategies for inbound and outbound throttling, see "Throttling condition triggers, actions, and mitigation strategies" in BizTalk Server Help at <http://go.microsoft.com/fwlink/?LinkId=154695>.
- To use a PassThrough send pipeline instead of the default XML send pipeline, see "Managing Send Ports Using BizTalk Explorer" in BizTalk Server Help at <http://go.microsoft.com/fwlink/?LinkId=154696>.
- For information about sizing the tracking database, see "Tracking Database Sizing Guidelines" in the BizTalk Server Help at <http://go.microsoft.com/fwlink/?LinkId=154677>.
- For information about sizing the MessageBox, BizTalkDTADb, and BAMPrimaryImport databases, see "Identifying Bottlenecks in the Database Tier" in the BizTalk Server Help at <http://go.microsoft.com/fwlink/?LinkId=154678>.
- For information about avoiding contention in the MessageBox database, see [How to Avoid Disk Contention](#) (<http://go.microsoft.com/fwlink/?LinkId=158809>) in the BizTalk Server Performance Optimization Guide.

Best Practices for Maintaining Performance

This topic points you to the best practices that you should follow when resolving performance issues with a BizTalk Server system.

For weekly performance checks	Checklist: Performing Weekly Performance Checks
For monthly performance checks	Checklist: Performing Monthly Performance Checks

See Also

[Maintaining Performance](#)

Configuring Batching to Improve Adapter Performance

The way an adapter processes a batch can have a significant effect on performance. Because there is a fixed delay associated with each transaction, you should try to minimize the number of transactions by combining more than one operation into a single batch.

If you are submitting messages to BizTalk Server in batches, do not limit the batch size based only on the message count. For example, if the batch size is two and the adapter gets four messages of size 4 KB, 8 KB, 1 MB, and 5 MB respectively, the first batch will be of size 12 KB, and the second batch will be of size 6 MB. Because the BizTalk Messaging Engine processes all messages in a single batch sequentially, the second batch in this example will be processed much more slowly than the first batch. The effect of this is reduced throughput.

To handle this problem, we recommend that you batch based on both the message count and the total number of bytes in the batch (that is, batch size in bytes). There is no optimal number for total bytes. However, in a normal processing scenario, if the batch size exceeds 1 MB, you will start encountering poor concurrency and throughput.

Adapters generally process messages of varying size in the production environment. The sizes of incoming messages are likely to vary significantly. As a result, always use message count and total bytes to build the batch.

How to Adjust the Message Size Threshold

This topic describes how to improve performance when processing large messages by adjusting the message size threshold and the *%temp%* environment variable.

The message size threshold determines which documents are buffered to the file system during mapping. Increase this value on a system with a large amount of available memory to improve throughput when processing large messages. Raising the threshold lowers the number of large messages that are buffered. Buffering documents to disk conserves memory at a cost to overall throughput.

The default message size threshold is 1 MB.

Prerequisites

To perform the procedure in this topic, you must be logged on as a member of the BizTalk Server Administrators group.

To adjust the message size threshold

1. Create a DWORD value named TransformThreshold at the following location in the BizTalk Server registry:

```
HKLM\Software\Microsoft\BizTalk  
Server\3.0\Administration\TransformThreshold
```
2. Enter a decimal value with the number of bytes to set the new threshold to. For example, to increase the message size threshold from the default of 1 MB to 2 MB, enter a decimal value of 2097152.
3. To improve performance when buffering large messages to the file system during mapping, change the setting for the *%temp%* environment variable to a non-system disk.

See Also

[Maintaining Performance](#)

How to Adjust the Configuration Cache Refresh Interval

The configuration cache refresh interval defines the time period in which BizTalk Server updates the configuration of the endpoints. When you start BizTalk Server, all items in BizTalk Server administration, such as MessageBox databases, server properties, adapters, and connections to the Tracking database, are stored in the configuration cache. All items in the cache are refreshed by the configuration refresh interval. By default this is every 60 seconds, except for the server database connections and server properties. This means that if you change the general properties for a BizTalk group, such as the SMTP host, the changes are picked up within 60 seconds. System changes made outside the currently open instance of the BizTalk Server Administration console are not reflected until you refresh it.

The configuration cache refresh interval is part of the BizTalk group properties.

Prerequisites

To perform the procedure in this topic, you must be logged on as a member of the BizTalk Server Administrators group.

► To adjust the cache refresh interval

1. Click **Start**, click **All Programs**, click **Microsoft BizTalk Server 2010**, and then click **BizTalk Server Administration**.
2. In the console tree, expand **BizTalk Server 2010 Administration**, right-click **BizTalk Group**, and then click **Settings**.
3. In the **BizTalk Settings Dashboard** dialog box, select the **General** tab. For the **Configuration refresh interval** property, type or select the time (in seconds) that all items in the BizTalk Server administration cache must wait between configuration cache refreshes, and then click **OK**.



Note

Items involved in the refresh include the MessageBox databases, server properties, adapters, and connections to the Tracking database.



Note

By default, all objects in the configuration cache are refreshed every 60 seconds, except for the server database connections and server properties.

How to Disable Tracking

This topic describes how to disable tracking using the BizTalk Server Administration console. You can configure various tracking options during runtime for orchestrations, send ports, receive ports, and pipelines using the BizTalk Server Administration console. You can change the tracking options for an item at any time, without interrupting the business process.

Prerequisites

To perform the procedure in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators group. For more detailed information about permissions, see [Permissions for Managing an Application](#).



Note

If you only want to view the tracking options, you can be logged on as a member of BizTalk Server Operators group.

▶ To disable tracking for an orchestration

1. Click **Start**, click **All Programs**, click **Microsoft BizTalk Server 2010**, and then click **BizTalk Server Administration**.
2. In the console tree, expand **BizTalk Server 2010 Administration**, expand the BizTalk group, expand **Applications**, and then expand the application containing the orchestration for which you want to configure tracking.
3. Click **Orchestrations**, and from the right pane, right-click the orchestration for which you want to configure tracking, and then click **Properties**.
4. Click the **Tracking** tab, disable tracking options as described in the following table, and then click **OK**.

Use this	To do this
Track Events - Orchestration start and end	Clear this check box to disable tracking of the orchestration instance before and after processing of the entire business process.
Track Events - Message send and receive	Clear this check box to disable tracking of message send and receive events. This check box is available only if you select the Orchestration start and end check box.
Track Events - Shape start and end	Clear this check box when you do not need to debug orchestration instances in the Orchestration Debugger. When this check box is selected, the event list in the

	Orchestration Debugger is populated. This check box is available only if you select the Orchestration start and end check box.
Track Message Bodies - Before orchestration processing	Clear this check box to disable the saving and tracking of the actual message content prior to processing by the orchestration instance. This check box is available only if you select the Message send and receive check box.
Track Message Bodies - After orchestration processing	Clear this check box to disable the saving and tracking of the actual message content after processing by the orchestration instance. This check box is available only if you select the Message send and receive check box.
Track Message Properties - Incoming messages	Clear this check box to disable tracking of the promoted properties of an inbound message. This check box is available only if you select the Message send and receive check box.
Track Message Properties - Outgoing messages	Clear this check box to disable tracking of the promoted properties of an outbound message. This check box is available only if you select the Message send and receive check box.

► **To disable tracking for a send port**

1. Click **Start**, click **All Programs**, click **Microsoft BizTalk Server 2010**, and then click **BizTalk Server Administration**.
2. In the console tree, expand **BizTalk Server 2010 Administration**, expand the BizTalk group, expand **Applications**, and then expand the application containing the send port for which you want to configure tracking.
3. Click **Send Ports**, right-click the send port, click **Properties**, and then click **Tracking**.
4. Disable tracking options as described in the following table, and then click **OK**.

Use this	To do this
Track Message Bodies - Request message before port processing	Clear this check box to disable saving and tracking of message content before the

	<p>message is received.</p> <p>nNote You must enable message body pipeline tracking to successfully track the response message before port processing.</p>
Track Message Bodies - Request message after port processing	Clear this check box to disable saving and tracking of message content after the message is received.
Track Message Bodies - Response message before port processing	Clear this check box to disable saving and tracking of message content before the message is sent. This check box is available only for solicit-response send ports.
Track Message Bodies - Response message after port processing	Clear this check box to disable saving and tracking of message content after the message is sent. This check box is available only for solicit-response send ports.
Track Message Properties - Request message before port processing	Clear this check box to disable tracking of the promoted properties of an inbound message.
Track Message Properties - Request message after port processing	Clear this check box if you do not want to track the promoted properties of an outbound message.
Track Message Properties - Response message before port processing	Clear this check box to disable saving and tracking of message properties before the message is sent. This check box is available only for solicit-response send ports.
Track Message Properties - Response message after port processing	Clear this check box to disable the saving and tracking of properties after the message is sent. This check box is available only for solicit-response send ports.

► **To disable tracking for a receive port**

1. Click **Start**, click **All Programs**, click **Microsoft BizTalk Server 2010**, and then click

BizTalk Server Administration.

2. In the console tree, expand **BizTalk Server 2010 Administration**, expand the BizTalk group, expand **Applications**, and then expand the application containing the receive port for which you want to configure tracking.
3. Click **Receive Ports**, right-click the receive port, and then click **Tracking**.
4. Disable tracking options as described in the following table, and then click **OK**.

Use this	To do this
Track Message Bodies - Request message before port processing	Clear this check box to disable saving and tracking of message content before the message is received.
Track Message Bodies - Request message after port processing	Clear this check box to disable saving and tracking of message content after the message is received.
Track Message Bodies - Response message before port processing	<p>Clear this check box to disable saving and tracking of message content before the message is sent. This check box is available only for request-response receive ports.</p> <p>nNote You must enable message body pipeline tracking to successfully track the response message before port processing.</p>
Track Message Bodies - Response message after port processing	Clear this check box to disable saving and tracking of message content after the message is sent. This check box is available only for request-response receive ports.
Track Message Properties - Request message before port processing	Clear this check box to disable tracking of the promoted properties of an inbound message.
Track Message Properties - Request message after port processing	Clear this check box if you do not want to track the promoted properties of an outbound message.
Track Message Properties - Response message before port processing	Clear this check box to disable saving and tracking of message properties before the message is sent. This check box is

	available only for request-response receive ports.
Track Message Properties - Response message after port processing	Clear this check box to disable the saving and tracking of properties after the message is sent. This check box is available only for request-response receive ports.

► **To disable tracking for a policy**

1. Click **Start**, click **Programs**, click **Microsoft BizTalk Server 2010**, and then click **BizTalk Server Administration**.
2. In the console tree, expand **BizTalk Server 2010 Administration**, expand the BizTalk group, expand **Applications**, and then expand the application containing the policy for which you want to configure tracking.
3. Click **Policies**, right-click the policy, click **Properties**, and then click **Tracking**.
4. Disable tracking options as described in the following table, and then click **OK**.

Use this	To do this
Fast activity	Clear this check box to disable tracking of the instance data on which the policy operates.
Condition evaluation	Clear this check box to disable tracking of the true/false results of conditions in the selected policy.
Rule firings	Clear this check box to disable tracking of the actions started as a result of the policy.
Agenda updates	Clear this check box to disable tracking of updates to the agenda. The agenda contains a list of actions that are "true" and need to fire.

► **To disable tracking for a schema**

1. Click **Start**, click **Programs**, click **Microsoft BizTalk Server 2010**, and then click **BizTalk Server Administration**.
2. In the console tree, expand **BizTalk Server 2010 Administration**, expand the BizTalk group, expand **Applications**, and then expand the application containing the schema for which you want to configure tracking.
3. Click **Schemas**, right-click the schema, and then click **Properties**.

4. In the left pane, click **Tracking**.
5. Do one of the following to specify which properties that you want to disable for tracking messages, and then click **OK**:
 - Clear the **Always track all properties** check box if you do not want to use all message properties regardless of the schema version. This check box is available only for document schemas.
 - Clear the **Select all message properties** check box if you do not want to use all the listed properties.
 - Under **Properties list**, clear the check box of each property that you do not want to use.



Note

You should select only the options that you need, because tracking messages creates performance and storage overhead for your system.

► **To disable tracking for a pipeline**

1. Click **Start**, click **Programs**, click **Microsoft BizTalk Server 2010**, and then click **BizTalk Server Administration**.



Note

If you set tracking options on pipelines, you will also set the tracking options globally for every port that uses the pipeline. This, in turn, may result in far more data being tracked than you intended, which will slow system performance. Instead, you can set tracking options on send ports and receive ports.

2. In the console tree, expand **BizTalk Server 2010 Administration**, expand the BizTalk group, expand **Applications**, and then expand the application containing the pipeline for which you want to configure tracking.
3. Do one of the following:
 - To disable tracking for one of the default BizTalk pipelines, expand <All Artifacts>.
 - To disable tracking for a custom pipeline that has been deployed into a BizTalk application, expand the application containing the pipeline.
4. Click the **Pipelines** folder, right-click the pipeline, and then click **Tracking**.

Note

To disable tracking for multiple pipelines at once, hold down the Shift key, click each pipeline to configure, right-click one of the pipelines, and then click **Tracking**.

5. Disable tracking options as described in the following table, and then click **OK**.

Use this	To do this
Port start and end events	Clear this check box to disable tracking only when an instance starts and ends.

	Details include item name, assembly, and other metadata.
Message send and receive events	Clear this check box to disable the tracking of message send and receive events. This check box is available only if Port start and end events is selected.
Message before pipeline processing	<p>Clear this check box to disable saving and tracking of the message bodies received by the pipeline, which holds metadata such as URLs and promoted properties. If this is a receive pipeline, the message body is the raw message as submitted to the pipeline by the transport component. Depending on the application, the message might be encrypted, signed, or encoded. When using a BizTalk Server map, if this is a receive pipeline, tracking takes place after the inbound map is processed.</p> <p>This check box is available only if Message send and receive events is selected.</p>
Message after pipeline processing	<p>Clear this check box to disable saving and tracking of the message bodies sent by the pipeline, which holds metadata such as URLs and promoted properties. If this is a receive pipeline, the message body is the processed message to be submitted to the MessageBox database, which may be XML depending on your application. When using a BizTalk Server map, if this is a send pipeline, tracking takes place before the outbound map is processed.</p> <p>This check box is available only if Message send and receive events is selected.</p>

See Also

[Maintaining Performance](#)

Troubleshooting Performance Issues

This section contains general guidelines for diagnosing and resolving performance problems related to BizTalk Server and its dependencies. These guidelines may also be used preemptively, to identify potential problems before they become critical issues.

Diagnosing Performance Problems in the BizTalk Server Environment

Typically a performance problem can be narrowed down to one of the following components of a BizTalk Server environment:

- A receive adapter or the system from which the adapter is receiving documents. For example, if documents are being received by the HTTP adapter at a suboptimal rate then the problem may be with the HTTP receive adapter or with the client that is posting to the HTTP adapter.
- An orchestration service instance.
- Performance of the SQL Server that hosts the BizTalk Server databases.
- A send adapter or the system that the adapter is sending documents to. For example, if documents are being sent by the SQL adapter at a suboptimal rate then the problem may be with the SQL send adapter or with the computer running SQL Server that the SQL adapter is updating.

Use the following guidelines to help identify the components of the BizTalk Server environment that are performing poorly:

- Capture any warnings or errors generated in the BizTalk Server or SQL Server Event Viewer.
- Follow the steps in "Identifying Performance Bottlenecks" in BizTalk Server Help at <http://go.microsoft.com/fwlink/?LinkId=154238> to help identify performance bottlenecks.

Once the poorly performing component has been identified, follow the appropriate guidelines to help resolve the issue:

Guidelines for resolving performance problems related to send and receive adapters

- For information about troubleshooting problems with BizTalk Server adapters, see "Troubleshooting BizTalk Server Adapters" in BizTalk Server Help at <http://go.microsoft.com/fwlink/?LinkId=154240>. This section contains general troubleshooting information including information about how to set up logging for certain adapters, and information that can be used to diagnose network problems, problems with MSDTC, problems with the registry, problems with the file system, and problems with IIS.
- For information about troubleshooting problems with MSDTC, certificates, Enterprise Single Sign-On, and SQL Server, see the appropriate section of "Troubleshooting BizTalk Server Dependencies" in BizTalk Server Help at <http://go.microsoft.com/fwlink/?LinkId=154242>.

Guidelines for resolving performance problems related to orchestrations

- For information about modifying the appropriate sections of the BTSNTSvc.exe.config file, see "Orchestration Engine Configuration" in BizTalk Server Help at <http://go.microsoft.com/fwlink/?LinkId=154244>.

Guidelines for resolving performance problems related to SQL Server

- SQL Server Profiler can be used to capture Transact-SQL statements that are sent to SQL Server and the SQL Server result sets from these statements. Since BizTalk Server is tightly integrated with SQL Server, the analysis of a SQL Server Profile trace can be a useful tool for analyzing problems that may occur in BizTalk Server when reading from and writing to SQL Server databases. For information about how to use SQL Server Profiler, see "Using SQL Server Profiler" in SQL Server Books Online at <http://go.microsoft.com/fwlink/?linkid=104423>.
- SQL Server Management Studio can be used to execute SQL statements directly against SQL Server databases. This functionality may be useful for querying the BizTalk Server databases or for updating the BizTalk Server databases in certain scenarios. For more information about using SQL Server Management Studio to execute SQL statements, see "Writing, Analyzing, and Editing Scripts with SQL Server Management Studio" in SQL Server Books Online at <http://go.microsoft.com/fwlink/?linkid=104425>.
- For more information about resolving performance problem related to the BizTalk Server databases, see "Troubleshooting SQL Server" in BizTalk Server Help at <http://go.microsoft.com/fwlink/?LinkId=154250>.

See Also

<http://go.microsoft.com/fwlink/?linkid=104430>

Maintaining BizTalk Server Databases

For information about maintaining BizTalk Server databases, see [Checklist: Maintaining and Troubleshooting BizTalk Server Databases](#).

See Also

[Maintaining BizTalk Server](#)

Increasing Availability for BizTalk Server

This section describes ways you can increase the availability of your BizTalk Server system.

Strategies for Increasing Availability

Strategies for increasing availability include the following:

- **Providing high availability using Windows Server 2003 server clustering or Windows Server 2008 failover clustering.** A server/failover cluster is a group of independent computer systems, known as nodes, working together as a single system to ensure that critical applications and resources remain available to clients. If one of the nodes becomes unavailable as a result of failure or maintenance downtime requirements, another node immediately begins providing service (a process known as failover).
 - A server/failover cluster is typically recommended for the computers running SQL Server that house the BizTalk Server databases.
 - A server cluster may be required to provide high availability for certain BizTalk adapters.
 - A server cluster is typically recommended for the Enterprise Single Sign-On master secret server.
- **Providing high availability using a form of load balancing.**
 - Network load balancing (NLB). NLB delivers high availability by redirecting incoming network traffic to working NLB cluster hosts if a host fails or is offline. Unlike server clusters, NLB does not require special hardware.
 - BizTalk host load balancing. BizTalk host load balancing is provided for BizTalk Hosts by adding multiple servers running BizTalk Server to a BizTalk Server group and then configuring multiple instances of an in-process host to run on these servers. This distributes the execution of services and artifacts configured in that host across multiple instances of the host, which enhances its availability and scalability.



Note

Host load balancing functionality is available for in-process hosts only.

- Load balancing is provided for SQL Server disks through the use of a SAN or by adding multiple MessageBox databases.
- Strategies for providing **increased availability**. These strategies provide increased availability but usually also require that an administrator perform one or more actions at runtime. Therefore these strategies are typically thought of as providing increased availability as opposed to high availability:
 - Increasing availability using BizTalk Server log shipping and disaster recovery.
 - Increasing availability through implementation of the appropriate monitoring and maintenance strategies.

Difference Between Clustering and Disaster Recovery

While clusters and disaster recovery both increase availability, a key difference between them is that clusters typically provide much faster recovery time than disaster recovery does. Therefore a solution built on server/failover clusters or load balancing is commonly thought of as providing high availability as opposed to merely providing availability.

Disaster recovery allows you to resume operation of a failed system but is typically a manual process and requires more recovery time than a high-availability implementation. Therefore, a disaster recovery implementation provides availability but not high availability. You should employ

both high availability through server/failover clusters and load balancing, and availability through disaster recovery, in a production BizTalk Server environment.

In This Section

- [Providing High Availability](#)
- [Disaster Recovery](#)

See Also

[Checklist: Providing High Availability with Fault Tolerance or Load Balancing](#)

[Checklist: Increasing Availability with Disaster Recovery](#)

Providing High Availability

High availability is provided for a BizTalk Server environment by implementing redundancy for each functional component in the environment. Redundancy for a BizTalk Host can be accomplished by installing multiple computers running BizTalk Server in a BizTalk group and then configuring instances of the host to run on more than one of the computers running BizTalk Server in the group. Redundancy for other components in a BizTalk Server environment can be accomplished by configuring the components as resources in a Windows Server cluster. BizTalk Server also accommodates configuring BizTalk Hosts as resources in a Windows Server cluster, which is recommended to provide high availability for certain BizTalk adapters.

This section describes how to provide high availability for the functional components in a BizTalk Server environment.

In This Section

- [Planning for High Availability](#)
- [High Availability for BizTalk Hosts](#)
- [High Availability for Databases](#)
- [High Availability for the Master Secret Server](#)

See Also

[Disaster Recovery](#)

Planning for High Availability

High availability for BizTalk Server 2010 focuses on recovering functional components that might disrupt availability in a BizTalk Server deployment.

To demonstrate high availability in BizTalk Server 2010, you have to cause a failure and measure the product's effectiveness in recovery. A highly available BizTalk Server deployment makes errors and failures transparent to external applications and systems, and ensures that all services continue functioning correctly with minimal disruption.

Designing a BizTalk Server 2010 deployment that provides high availability involves implementing redundancy for each functional component involved in an application integration or business process integration scenario. BizTalk Server 2010 simplifies the implementation of these scenarios by conceptually separating the data from the *hosts* that process the data. A *host* is a logical container of BizTalk items, such as orchestrations, send handlers, and receive handlers. You create *host instances* and assign them to the host. A host instance is the physical representation of a host on a specific server. It is either the BizTalk Server 2010 service process called BTSNTSvc.exe or another process, for example, the IIS process. So providing high availability for BizTalk Server 2010 involves running multiple *host instances* and clustering the BizTalk Server databases, as follows:

- **Architecture for BizTalk Hosts.** BizTalk Server 2010 lets you separate hosts and run multiple host instances to provide high availability for key functions such as receiving messages, processing orchestrations, and sending messages. These hosts do not require any additional clustering or load-balancing mechanism because BizTalk Server 2010 automatically distributes workload across multiple computers through host instances. However, hosts running the receive handlers for the HTTP and SOAP adapters require a load-balancing mechanism such as Network Load Balancing (NLB) to provide high availability, and hosts running the receive handlers for FTP, MSMQ, POP3, SQL, and SAP require a clustering mechanism to provide high availability.

 **Note**

You must always cluster the SAP receive adapter to accommodate a two-phase commit scenario.

- **Architecture for BizTalk Server databases.** High-availability configuration for the BizTalk Server 2010 databases typically consists of two or more SQL Server database computers configured in an active/passive server cluster configuration. These computers share a common disk resource (such as a RAID 1+0 SCSI disk array or a storage area network) and use Windows Failover Clustering to provide backup redundancy and fault tolerance.

Another BizTalk functional component that is critical for high availability is the master secret server. BizTalk Server relies on this service to obtain the encryption key.

This section provides information about how to address high availability in each of these categories. Because a BizTalk Server 2010 high availability solution is built on Windows Server 2008 SP2 and SQL Server 2008 SP1, make sure that you deploy these products with high availability before configuring hosts for BizTalk Server 2010. The following links include information about providing high availability for these underlying products:

- **White Paper: High Availability—Always On Technologies** available at <http://go.microsoft.com/fwlink/?LinkId=156812>.
- Get more information about troubleshooting issues with Windows Server 2008 at the [Windows Server 2008 Deployment, Managing and Troubleshooting page](http://go.microsoft.com/fwlink/?LinkId=156813) (<http://go.microsoft.com/fwlink/?LinkId=156813>).
- Get more information about availability and scalability in Windows Server 2008 at [Availability and Scalability](http://go.microsoft.com/fwlink/?LinkId=156814) (<http://go.microsoft.com/fwlink/?LinkId=156814>).
- See the **High Availability** section of the [Windows Server 2008 R2 Articles and Whitepapers page](http://go.microsoft.com/fwlink/?LinkId=157760) (<http://go.microsoft.com/fwlink/?LinkId=157760>).

Understanding the Impact of a Component Failure

The following table lists the components and dependencies of a BizTalk Server environment and the impact on the BizTalk Server environment if the component or dependency fails. You should consider the scope of a potential failure when deciding whether to cluster a component or dependency.

Component or dependency	Scope of failure
SQL Server	Systemwide. If SQL Server fails then BizTalk Server will be unable to process documents.
Master secret server	<p>Systemwide. If the master secret server fails then BizTalk Server will be unable to process documents.</p> <p> Note If the master secret server fails, each BizTalk server in the BizTalk group will continue to use a cached in-memory copy of the master secret until the Enterprise SSO service on that BizTalk server is restarted. If the Enterprise SSO service is restarted on the BizTalk servers, then the cached copy of the master secret is released from memory and the BizTalk servers must be able to contact the master secret server to obtain another copy of the master secret. Do not restart the Enterprise SSO service on the BizTalk server(s) in a group if the master secret server fails and you want the BizTalk server to continue processing documents.</p>

Component or dependency	Scope of failure
MSDTC	<p>Server. If MSDTC fails then any component on the server that requires transaction support will fail.</p> <p> Note Because SQL Server and the master secret server are dependent on MSDTC for transaction support, the scope of the failure will become system wide if the MSDTC on the SQL server or master secret server fails. BizTalk Server requires transaction support when communicating with SQL Server and the master secret server during run-time operations.</p>
BizTalk Host instance	Server. Any components housed in a BizTalk Host instance will be unable to participate in document processing if the host instance fails.
Microsoft Message Queuing (MSMQ)	Server. If MSMQ fails then any document processing that is dependent on the MSMQ service, such as the MSMQ adapter, will be halted on the server.
File system	Server. If the file system fails then any document processing that is dependent on the file system, such as the File adapter, will be halted on the server.

To be able to better manage a highly available BizTalk Server system, you must have a good understanding of the BizTalk stack: Windows Server, DC (DNS, DHCP), BizTalk Server, SQL Server, IIS server, File server, MSMQ server, external applications. This section focuses on the high availability of BizTalk Server and the dependent SQL Server computer.

BizTalk Server High-Availability Examples

For sample scenarios in Microsoft BizTalk Server that provide high availability through scaled-out tiers of hosts, see [Sample BizTalk Server High Availability Scenarios](http://go.microsoft.com/fwlink/?LinkId=156815) (<http://go.microsoft.com/fwlink/?LinkId=156815>).

See Also

[High Availability for BizTalk Hosts](#)

[High Availability for Databases](#)

[High Availability for the Master Secret Server](#)

[Checklist: Increasing Availability with Disaster Recovery](#)

High Availability for BizTalk Hosts

BizTalk Server 2010 provides great flexibility in addressing high availability because you can strategically dedicate logical hosts to run specific areas of functionality, such as receiving and sending messages or processing orchestrations, that can be physically deployed to multiple servers.

A BizTalk Host is a logical container within a BizTalk Server group that can house BizTalk Server items such as adapter send handlers (including pipelines), receive locations, and orchestrations. You typically group items that have similar scale properties into a particular host.

After you create a host, you can deploy it to a physical BizTalk Server computer as a host instance. A host instance runs as a Windows service, `BTSNTSvc.exe` (or `BTSNTSvc64.exe` for 64-bit host instance), on the designated BizTalk Server computer. For each host, you can have only one instance on a particular BizTalk Server computer. However, you can have instances of a particular host on one or more BizTalk Server computers, and you can have instances of different hosts on a particular BizTalk Server computer.

Items that are contained in BizTalk Hosts can perform the following functions:

- **Receiving.** These items do the initial processing of messages after they are picked up in a receive location. When a host contains a receiving item, such as a receive location (with a pipeline), the message decoding and decrypting occurs in a pipeline within the host.
- **Sending.** These items do the final processing of messages before they are sent out to the send port. When a host contains a sending item, such as a send port, the message signing and encryption occurs in a pipeline within the host.
- **Processing.** These items process messages based on the instructions in orchestrations.

One BizTalk Host can contain items that receive, send, and process messages. For easier management and scalability, we recommend that you create different hosts designated for each function. In particular, we recommend that you use different hosts for processing and for receive/send operations.

For example, if you receive one message, run an orchestration, and send ten messages, you want to separate the receive and send functionality into two separate hosts because the send items will have ten times more traffic than the receive items. If you receive one message, run an orchestration, and send one message, you can think of these items as a unit of work and group them into one single host. Alternatively, you can separate them into three different hosts to increase performance and flexibility, although this also increases the management cost.

BizTalk Hosts are one of two types, *In-process* or *Isolated*. In-process hosts run inside of the BizTalk Server runtime process (BTSNTSvc.exe or BTSNTSvc64.exe) and Isolated hosts do not run in the BizTalk Server runtime process. Isolated hosts are only used on the receiving side for the isolated receive adapters. The following table lists the items that each of these host types may contain.

Host type	Logical container for
In-process	<ul style="list-style-type: none"> • Orchestrations • Adapter send handlers • In-process adapter receive handlers
Isolated	<ul style="list-style-type: none"> • HTTP, SOAP receive handlers • Any other isolated adapter receive handlers

For more information about managing BizTalk Hosts and host instances, see [Managing BizTalk Hosts and Host Instances](http://go.microsoft.com/fwlink/?LinkID=154191) (http://go.microsoft.com/fwlink/?LinkID=154191) in BizTalk Server 2010 Help.

To provide high availability for BizTalk Hosts, you must have two or more host instances for each host (on two or more computers) in your environment. By having more than one host instance for each host you make sure that if one host instance becomes unavailable, the host instances on other computers that are running instances of the same host can resume the functions of the problematic or failed host instance, and that the overall system can continue performing with minimal disruption.

Disadvantages of Additional Hosts

While there are benefits of creating additional host instances, there are also potential drawbacks if too many host instances are created. Each host instance is a Windows service (BTSNTSvc.exe or BTSNTSvc64.exe), which generates additional load against the MessageBox database and consumes computer resources, such as CPU, memory, and threads. Other than these, you have the following reasons for not configuring too many additional host instances:

- Several performance counters are reported per host with too much granularity. This affects the usability for the administrator who would need to traverse through a lot of data. This has a negative impact on the overall view the administrator has.
- Each host consumes considerable amount of memory that might lead to a situation of throttling and reduced performance.
- If the hosts have receive adapters that continuously perform polling, each host will poll the database at short intervals, thereby resulting in degraded performance.

In This Section

- [Scaling Out Receiving Hosts](#)

- [Clustering Receiving Hosts](#)
- [Scaling Out Processing Hosts](#)
- [Scaling Out Sending Hosts](#)

See Also

[Configuring Hosts and Host Instances](#)

[Configuring a Dedicated Tracking Host](#)

[Planning for High Availability](#)

[High Availability for Databases](#)

[High Availability for the Master Secret Server](#)

Scaling Out Receiving Hosts

To make receiving hosts highly available, you must have two or more BizTalk Server computers that are running instances of each receiving host. By scaling out the receiving hosts you can increase the availability for BizTalk Server 2010 deployments that are messaging-intensive. While these deployments might perform minimal processing for orchestrations, they can route many messages of varying types with great speed and reliability.

You can enhance security and scalability in your environment by separating the receiving host from the hosts that process orchestrations and send messages because you can secure and scale each host independently of other hosts. For example, you can add two computers (host instances) to the receiving host without adding any computers to the processing or sending hosts.

Understanding In-Process and Isolated Receiving Hosts

BizTalk Server integrates applications to provide business services. The integration is usually represented as BizTalk Server receives a document (from an application), processes the document, and sends the processed document back to the application or to another application. The process is called a document transaction.

A transaction usually starts with a BizTalk adapter monitoring a certain protocol channel and receiving a document. The *adapter* is so called because it connects other applications to BizTalk Server. Based on its function, it can be a send adapter or a receive adapter. Most of the default adapters are one .NET component with both the receive function and send function built into one .NET assembly. Based on the process memory space in which an adapter resides, it is either an in-process (receive) adapter or an isolated (receive) adapter. An in-process adapter can only be hosted by the BizTalk Server process (BTSNTSvc.exe), and an isolated adapter is designed to be hosted by another process. For example, the HTTP adapter and the SOAP adapter are hosted by

the Internet Information Services (IIS) process. They are essentially ISAPI extensions. On the other hand, all of the send adapters are in-process adapters.

BizTalk Server Configuration creates two default hosts—the in-process host is called BizTalkServerApplication, and the isolated host is called BizTalkServerIsolatedHost. A host serves two functions: one is logically grouping the BizTalk Server items so these items can be assigned to different BizTalk Server processes, and the other is controlling security. You need to specify a Windows group for a host. Only the users in this group can send documents to the adapters hosted by the *host instances* assigned to this host.

Each of the two default hosts has a host instance. A host instance does not have a name, but is associated with a host. The BizTalkServerApplication host instance is actually the BizTalk Server service process (BTSNTSvc.exe) on a BizTalk Server computer within the BizTalk group. The BizTalkServerIsolatedHost host instance is not directly bound to a process. It is associated with the process that hosts the receive adapter.

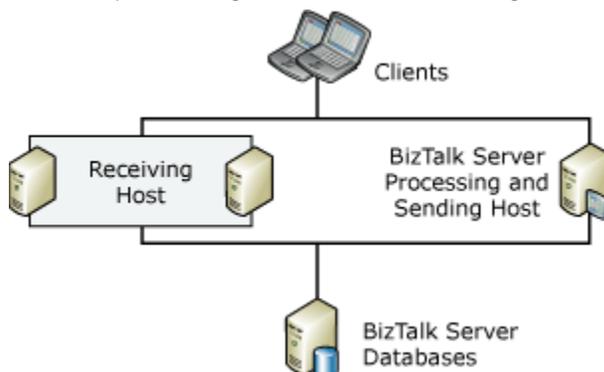
BizTalk Server Configuration also creates a *receive handler* for each of the default adapters with the exception of SMTP (SMTP is a send adapter). One of the receive handler properties is host name. That is how it is bound to a host and the host instances of that host.

In addition to an adapter, host, host instance, and receive handler, you need to configure a receive port before BizTalk Server can start receiving documents. A receive port contains receive locations. A receive location has a receive handler property. Following the logic, you can trace to the BizTalk Server process that processes this receive port.

In the receive port configuration, you optionally specify maps. In the receive location configuration, you must specify a pipeline used for document preprocessing. The designated BizTalk Server process handles everything from receiving a document, to preprocessing the document, to mapping the document. This is the same for both in-process host instances and isolated host instances.

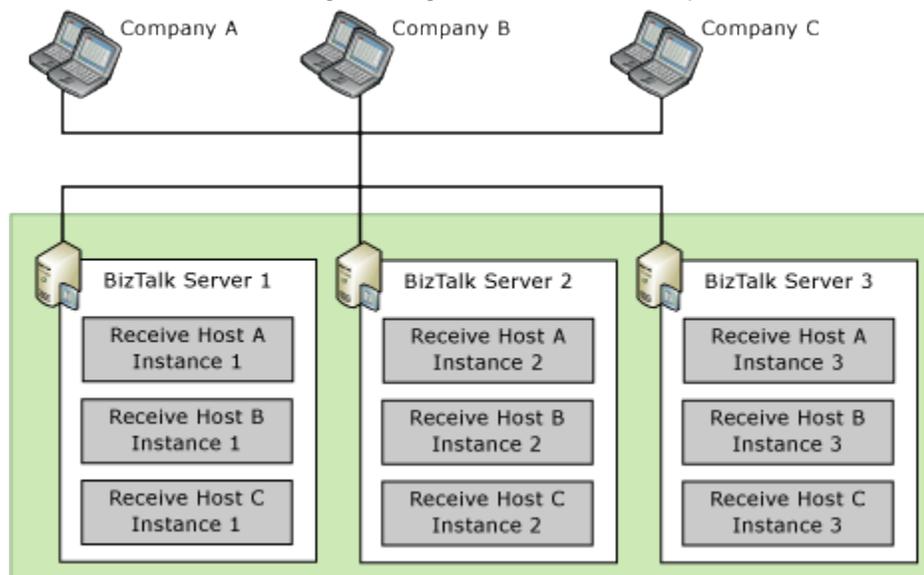
Scaling Out In-Process Receiving Hosts

The following figure shows a BizTalk Server 2010 deployment that provides high availability for the receiving host by having two host instances each on a different computer. Note that in this figure the processing and sending host is not highly available, because there is only one host instance processing the BizTalk items assigned to the host.



For large deployments, for scenarios dealing with multiple trading partners, and for scenarios when you use different protocols, you can spread out the receiving functionality across multiple receiving hosts. For example, you can create a host for receiving messages for each adapter, or different hosts for receiving messages from different partners. When you create multiple receiving hosts you can create security boundaries and ease the manageability and scalability of your environment; however, it does not make your environment highly available. To make your environment highly available, you must create two or more host instances for each receiving host that you create. For example, you can create three different receiving hosts (A, B, and C) to receive messages from three different companies. To make each of these hosts highly available you then create host instances of each of these hosts in two or more computers. Note that you can have instances of each host on one computer without losing the security boundary, manageability, or scalability.

The following figure shows a highly available three-computer BizTalk Server environment with hosts dedicated to receiving messages from different companies.



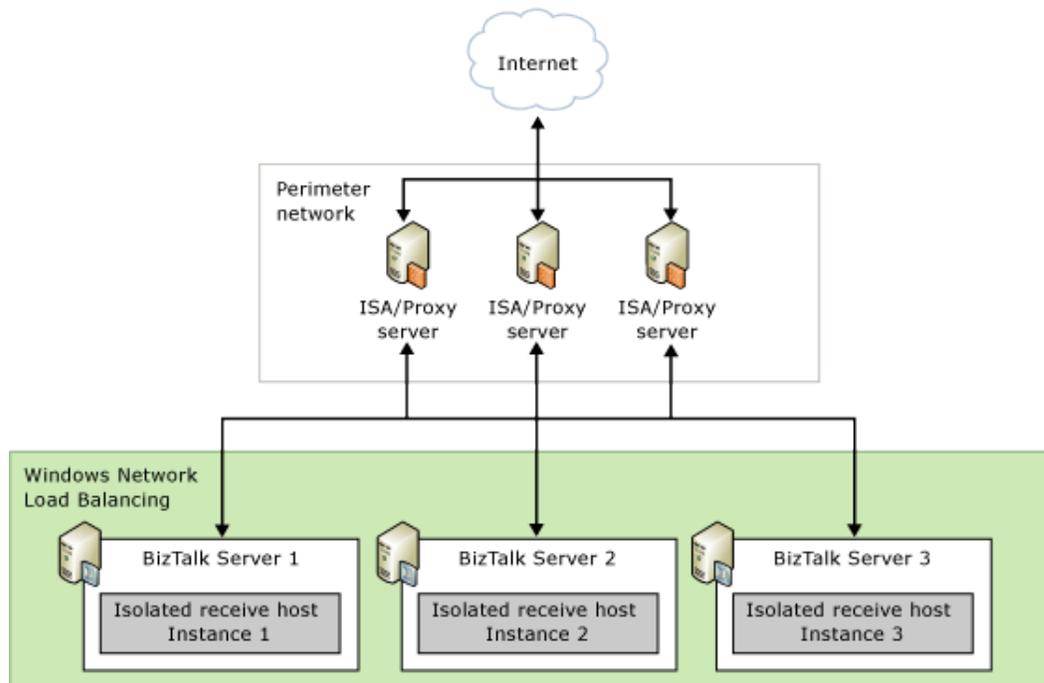
To provide high availability in this configuration, each computer runs three host instances: one instance for each of the three companies. The host instances for each company contain the receive locations and pipelines to communicate with that company. During typical operations, as long as you have done the necessary work for scale-out in front of the receive adapters, the messaging load is distributed among the three host instances for each host. If a host instance on one computer fails, the host instances running on the other two computers provide redundancy and maintain service availability.

Scaling Out Isolated Receiving Hosts

Besides host instances, the process of scaling and providing high availability for the receiving hosts also depends on the specific adapters that you implement in your deployment. Each adapter has protocol-specific characteristics that make planning and deployment different in each

case. However, BizTalk Server 2010 lets you apply the same high-availability solution for all adapters, primarily through additional computers and host instances.

Depending on the specific protocol being used, some receive adapters require an additional mechanism for distributing incoming messages among multiple host computers to provide high availability. For example, BizTalk Server 2010 solutions that use the HTTP or SOAP adapter (otherwise known as the Web services adapter) require a load balancer such as Network Load Balancing (NLB) to distribute the receiving workload, as shown in the following figure.



For more information about the high-availability guidelines for the most common adapters in BizTalk Server, see "Scaling the BizTalk Server Receive Adapters" section in [Scaled-Out Receiving Hosts](http://go.microsoft.com/fwlink/?LinkId=151283) (<http://go.microsoft.com/fwlink/?LinkId=151283>) in BizTalk Server 2010 Help.

See Also

[Clustering Receiving Hosts](#)

[Scaling Out Processing Hosts](#)

[Scaling Out Sending Hosts](#)

Clustering Receiving Hosts

BizTalk Server 2010 provides functionality that allows you to configure a BizTalk Host as a clustered resource within a Windows Server cluster group. Host cluster support is provided to

support high availability for integrated BizTalk receive adapters that should not be run in multiple host instances simultaneously, such as the FTP receive handler or, under certain circumstances, the POP3 receive handler. Host cluster support is also provided to ensure transactional consistency for messages sent or received by the MSMQ adapter in scenarios that require that the MSMQ service is clustered.



Note

Host clustering is available only with BizTalk Server 2010 Enterprise Edition.



Note

Before you can cluster a BizTalk Host you must have configured at least two BizTalk Server computers in a BizTalk group as members of a Windows Server cluster. For more information about configuring a Windows Server cluster, see [Windows Server 2008 Clustering Documents, Whitepapers, Webcasts, Groups](http://go.microsoft.com/fwlink/?LinkId=156818) (<http://go.microsoft.com/fwlink/?LinkId=156818>).

BizTalk Host cluster support is available to provide high availability for five of the integrated BizTalk adapters: the FTP adapter, the MSMQ adapter, the POP3 adapter, the SQL adapter, and the SAP adapter. Host cluster support is also provided so that there is high availability for running a single instance of an adapter for purposes of ordered delivery.

All of the BizTalk adapter handlers can be run in a clustered host, but there is no benefit from running adapter handlers in a clustered host except as described below. If your processing requirements include any of the scenarios described in the section below, then you should run adapter handlers in a clustered host. For detailed steps of setting up BizTalk Server clusters, see [Improving Fault Tolerance in BizTalk Server by Using a Windows Server 2008 Failover Cluster or Windows Server 2003 Server Cluster](http://go.microsoft.com/fwlink/?LinkId=156819) (<http://go.microsoft.com/fwlink/?LinkId=156819>).

Scenarios for Running Adapter Handlers in Clustered Hosts

If your processing requirements include any of the scenarios listed below, then you should run adapter handlers in a clustered host.

- Running the FTP adapter receive handler within a clustered BizTalk host
- Running MSMQ adapter handlers within a clustered BizTalk host
- Running the POP3 adapter receive handler within a clustered BizTalk host
- Running a receive adapter that supports ordered delivery with a clustered BizTalk host

For more information about these scenarios, see [Considerations for Running Adapter Handlers within a Clustered Host](http://go.microsoft.com/fwlink/?LinkId=151284) (<http://go.microsoft.com/fwlink/?LinkId=151284>).

See Also

[Scaling Out Receiving Hosts](#)

[Scaling Out Processing Hosts](#)

[Scaling Out Sending Hosts](#)

Scaling Out Processing Hosts

For information about scaling out processing hosts, see [Scaled-Out Processing Hosts](#) (<http://go.microsoft.com/fwlink/?LinkId=151285>).

See Also

[Scaling Out Receiving Hosts](#)

[Clustering Receiving Hosts](#)

[Scaling Out Sending Hosts](#)

Scaling Out Sending Hosts

For more information about scaling out sending hosts, see [Scaled-Out Sending Hosts](#) (<http://go.microsoft.com/fwlink/?LinkId=151286>).

See Also

[Scaling Out Receiving Hosts](#)

[Clustering Receiving Hosts](#)

[Scaling Out Processing Hosts](#)

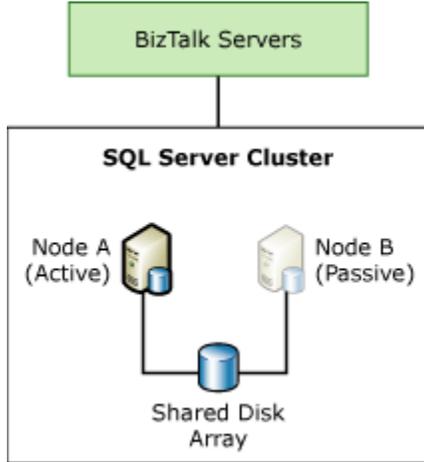
High Availability for Databases

BizTalk Server relies heavily on SQL Server for data store and data persistence. All other components and hosts in BizTalk Server have specific roles in the process of integrating disparate business applications (for example, receiving, processing, or routing messages), but the database computer captures this work and persists it to disk. For example, when BizTalk Server receives an incoming message, the receive host persists it to the MessageBox database before other hosts retrieve the message for orchestration processing and sending. If your BizTalk solution involves orchestration, BizTalk Server routes the message to the host that executes the business process (processing host), and saves the message to the MessageBox database after the orchestration finishes. The sending host then retrieves the message from the database before sending it to the external application through the appropriate send adapter.

To provide high availability for the BizTalk Server databases, use Windows Clustering to configure two or more computers that are running SQL Server to create a server cluster. This server clustering provides redundancy and fault tolerance for the BizTalk Server databases.

Unlike load-balanced clustering, where a group of computers functions together to increase availability and scalability, server clustering typically involves a pair of database computers in an active/passive configuration so that one computer provides backup resources for the other.

The following figure shows a BizTalk Server database tier with high availability through active/passive server clustering.



If the active database computer encounters errors or fails, the passive computer becomes active and assumes control over the database resources until the failed computer is repaired. The database service fails over and restores data connections to the new active computer and enables the BizTalk application to continue functioning.

BizTalk Server Databases

Microsoft BizTalk Server installs several databases in SQL Server. The following table shows typical usage characteristics for the BizTalk Server databases.

Database	Default database name	Usage characteristics
Management database	BizTalkMgmtDb	This database handles low-usage read and write operations.
MessageBox database	BizTalkMsgBoxDb	This database handles high-usage read and write operations.
Tracking database	BizTalkDTADb	This database handles potentially high-usage write operations depending on the amount of data that you configure to be tracked, and low-usage read operations.

Database	Default database name	Usage characteristics
SSO database	SSODB	This database handles low-usage read and write operations.
BAM Analysis database	BAMAnalysis	This SQL Server Analysis Services database handles potentially high-usage read and write operations, depending on the level of monitoring performed.
BAM Star Schema database	BAMStarSchema	This SQL Server Analysis Services database handles potentially high-usage read and write operations, depending on the level of monitoring performed.
BAM Primary Import database	BAMPrimaryImport	This SQL Server Analysis Services database handles potentially high-usage read and write operations, depending on the level of monitoring performed.
BAM Archive database	BAMArchive	This SQL Server Analysis Services database handles potentially high-usage read and write operations, depending on the level of monitoring performed.
Rule Engine database	BizTalkRuleEngineDb	This database handles potentially low-usage read and write operations, unless you update the rules.
Tracking Analysis Services database	BizTalkAnalysisDb	This SQL Server Analysis Services database handles high-usage read and write operations.

BizTalk Server runtime operations typically use the first four databases (Management database, MessageBox databases, Tracking database, and SSO database). Depending on the traffic on

these databases, you can put them on separate computers that are running SQL Server. Depending on the BizTalk Server functionality that you use, you may have some or all of the other databases in the table. You can scale out and cluster these databases as needed.

Make sure that you follow good SQL Server deployment practices, such as using separate disks for each database. For more information about SQL Server deployment best practices, see [White Paper: High Availability—Always On Technologies](#) (<http://go.microsoft.com/fwlink/?LinkId=156812>).

For the BizTalk Server databases, we recommend that you do the following:

- **Set up failover clustering.** Failover clustering enables SQL Server to automatically switch the processing for an instance of SQL Server from a failed server to a working server.
The BAM Primary Import database collects event data. In the event of a disaster, data that was written to the BAM Primary Import database since the last backup will be lost. Because there is no way to regenerate lost events, it is especially important that you enable failover clustering on your BAM Primary Import database.
- **Use SQL Server RAID 1+0 (redundant array of independent disks),** especially for the MessageBox database and the BAM Primary Import database.

For information about backing up your BizTalk Server databases, see [Best Practices for Disaster Recovery](#).



Note

Microsoft SQL Server 2008 R2 and SQL Server 2008 SP1 provide a software solution known as database mirroring for increasing the probability that a database is available. The use of SQL Server database mirroring is not currently a supported solution for ensuring high availability of the Microsoft BizTalk Server databases because of potential problems maintaining transactional consistency in the BizTalk Server databases.

For more information about database mirroring and cross-database transactions in SQL Server 2008 SP1, see <http://go.microsoft.com/fwlink/?LinkId=87977>. BizTalk Server databases should be installed on a SQL Server cluster to ensure high availability and log shipping should be utilized for purposes of disaster recovery.

For more information about log shipping, see [What Is BizTalk Server Log Shipping?](#)

In This Section

- [Clustering the BizTalk Server Databases](#)
- [Scaling Out the BizTalk Server Databases](#)

See Also

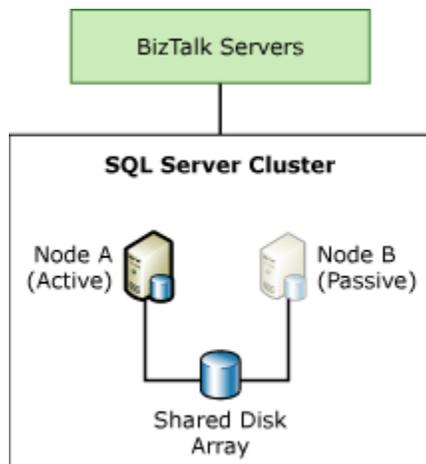
[Planning for High Availability](#)

[High Availability for BizTalk Hosts](#)

[High Availability for the Master Secret Server](#)

Clustering the BizTalk Server Databases

If the BizTalk Server databases become unavailable, the BizTalk Server environment will not function correctly. To provide high availability, you can create a Microsoft SQL Server cluster for the BizTalk Server databases, as shown in the following figure.



To create a highly available solution for the BizTalk Server databases, you must have at least two computers that are running SQL Server and a shared disk array in the cluster.

Clustering Options

Determine the best cluster configuration for the BizTalk Server databases for your business needs. Here is a list of the options:

- **Active/passive.** High availability for the BizTalk Server databases typically consists of two or more database computers configured in an active/passive server cluster configuration. These computers share a common disk resource (such as a RAID 1+0 SCSI disk array or storage area network) and use Windows Clustering to provide backup redundancy and fault tolerance.
- **Active/active.** Windows Clustering and SQL Server allow you to run SQL Server in Active/Active mode where each node of the cluster is “active” and running one or more SQL Server instances. For example, this would allow you to have the MessageBox database on one node and all other BizTalk Server databases on the other node. This allows you to maximize cluster hardware usage, but an active/active SQL Server configuration should be used with care.

Can each node simultaneously handle the load of all SQL Server instances during a SQL Server cluster node failover scenario? Are there enough CPU resources? Is there sufficient memory? What about network bandwidth? How about disk I/O contention?

These are just some of the questions that need to be answered in order to determine if an active/active SQL Server cluster is right for your BizTalk applications. If it is determined that one node cannot handle all SQL Server instances in a failover scenario, an alternative is to use active/active/passive clustering.

- **Active/active/passive.** The run-time processes write to the BizTalk Management database, MessageBox databases, Tracking Analysis Services database, BAM Analysis database, BAM Star Schema database, BAM Primary Import database, and BAM Archive database. Therefore, these databases are especially important if a disaster occurs, and must have greater priority when determining what databases to cluster. Only users or tools write to the other databases. For the MessageBox databases, you can consider an active/active/passive or active/active/active/passive configuration to minimize the hardware needed.



Note

SQL Server 2008 R2 and SQL Server 2008 SP1 Standard Edition supports 2-node failover clusters. If you decide to use the active/active/passive configuration on SQL Server 2008 R2 or SQL Server 2008 SP1, you must use the Enterprise Edition of SQL Server.

Procedures for Clustering the Databases

Make sure you meet the following prerequisites before you start clustering the BizTalk Server databases.

- When you create the domain groups for your BizTalk Server environment, you must create global domain accounts.
- Configure the SQL Server cluster before you install and configure BizTalk Server. For more information about clustering SQL Server 2008 R2, see [Getting Started with SQL Server 2008 R2 Failover Clustering](http://go.microsoft.com/fwlink/?LinkId=156820) (<http://go.microsoft.com/fwlink/?LinkId=156820>).
- If you are also clustering the master secret server, configure that server first. For more information about high availability for Enterprise Single Sign-On, see [High Availability for the Master Secret Server](#).

To run the BizTalk Server Configuration Wizard

1. Install BizTalk Server 2010 on a runtime server.
2. Launch the BizTalk Server Configuration program. Click **Start**, point to **Programs**, point to **Microsoft BizTalk Server 2010**, and then click **BizTalk Server Configuration**.
3. To apply a custom configuration, follow the steps in [Working with the Custom Configuration Manager](http://go.microsoft.com/fwlink/?LinkId=156822) (<http://go.microsoft.com/fwlink/?LinkId=156822>) in BizTalk Server Help. To specify the SQL Server cluster for the BizTalk Server databases enter the name of the SQL Server cluster in the **Databases** dialog of the configuration.
4. Complete the BizTalk Server configuration by following the instructions in [Custom Configuration](http://go.microsoft.com/fwlink/?LinkId=156823) (<http://go.microsoft.com/fwlink/?LinkId=156823>) in BizTalk Server Help.

For more information about clustering BizTalk Server databases, see [Improving Fault Tolerance in BizTalk Server by Using a Windows Server 2008 Failover Cluster or Windows Server 2003 Server Cluster](http://go.microsoft.com/fwlink/?LinkId=156819) (http://go.microsoft.com/fwlink/?LinkId=156819).

Behavior of BizTalk Host Instances During SQL Server Failover

For more information about behavior of BizTalk host instances during SQL Server failover, see [Behavior of BizTalk Server Host Instances during SQL Server Failover](http://go.microsoft.com/fwlink/?LinkId=151287) (http://go.microsoft.com/fwlink/?LinkId=151287).

Using SQL Server Database Mirroring

For more information about using SQL Server database mirroring with respect to BizTalk Server database clustering, see [Use of SQL Server database mirroring or the Volume Shadow Copy service](http://go.microsoft.com/fwlink/?LinkId=151288) (http://go.microsoft.com/fwlink/?LinkId=151288) in BizTalk Server help.

See Also

[Scaling Out the BizTalk Server Databases](#)

Scaling Out the BizTalk Server Databases

To provide high availability for the BizTalk Server databases, configure two computers that are running SQL Server 2008 SP1 or SQL Server 2005 in a Windows cluster. These computers can run in an active/active, active/passive, or active/active/passive (requires three computers) configuration for redundancy and can store data on a shared drive (such as a RAID 1+0 SCSI disk array) or storage area network (SAN).

If the SQL Server service becomes unavailable for any reason, the database cluster transfers resources from the active computer to the passive computer. During this failover process, the BizTalk Server service instances experience database connection failures and automatically restart to reconnect to the databases. The functioning database computer (previously the passive computer) begins processing the database connections after assuming the resources during failover.

Clustering the BizTalk Server databases is discussed in [Clustering the BizTalk Server Databases](#). This section focuses on scaling out the BizTalk Server databases to provide high availability.

Providing High Availability for the BizTalk MessageBox Database

This section provides information about to configure BizTalk MessageBox database for high availability.

Running Multiple MessageBox Databases

To enhance the scalability of the BizTalk Server databases and to address high CPU utilization on the MessageBox database SQL Server computer, you can configure BizTalk Server to store data across multiple MessageBox databases. You create the first MessageBox database when you run the Configuration Wizard. This MessageBox database is the master MessageBox database. There is a single master MessageBox database in your BizTalk Server deployment. The master MessageBox database contains the master subscription information and routes messages to the appropriate MessageBox database. Typically, you want to dedicate the master MessageBox database to do routing only and let the other MessageBox databases do the processing. To make a MessageBox database to do routing only, select **Disable new message publication** from the MessageBox properties in BizTalk Administration console.

An example of MessageBox database processing flow is as follows:

1. When the master MessageBox database receives a new activation message—a brand new instance of a business process or a subscription message—the master MessageBox database distributes the activation message to the next available MessageBox database. For example, if you have one master MessageBox database and two MessageBox databases, the master MessageBox database routes the first activation message to MessageBox database 1, the second activation message to MessageBox database 2, the third activation message to MessageBox database 1, and so on in a round-robin pattern. The master MessageBox database uses built-in logic to load balance, and does not need additional load-balancing mechanisms.
2. After the master MessageBox database routes the activation message to a particular MessageBox database (for example, MessageBox database 1), the business process goes into memory and runs.
3. If the business process has to wait for a message, and the wait time is longer than several seconds, the business process is persisted back into MessageBox database 1. The business process is waiting for a correlation message.
4. When the correlation message arrives at the master MessageBox database, the Message Engine does a lookup operation in the database for the MessageBox database that contains the state for the correlation message (in this example, MessageBox 1). The master MessageBox database delivers the message to the MessageBox database that contains the business process.
5. The business process is brought back into memory to continue processing until it finishes or until it has to wait for another correlation message.

BizTalk Server stores all the states in the MessageBox databases, and each MessageBox database contains state information for different business processes. For reliability, you must

cluster all the MessageBox databases, including the master and secondary MessageBox databases.

To configure multiple MessageBox databases, you use the BizTalk Server Administration console to add the computers that are running SQL Server 2008 R2 or SQL Server 2008 SP1. From an administration perspective, you only have to add new MessageBox databases. BizTalk Server automatically handles the round-robin distribution of activation messages and sends correlation messages to the correct MessageBox databases.

If you configure multiple MessageBox databases in your environment you should create a minimum of three MessageBox databases for your BizTalk Server group and you should disable message publication on the master MessageBox database. This recommendation is made because adding additional MessageBox databases incurs overhead by the master MessageBox database for routing messages between the MessageBox databases. If you only configure two MessageBox databases then most of the benefit gained by the additional MessageBox database is offset by the overhead consumed by the master MessageBox database for message routing.

 **Important**

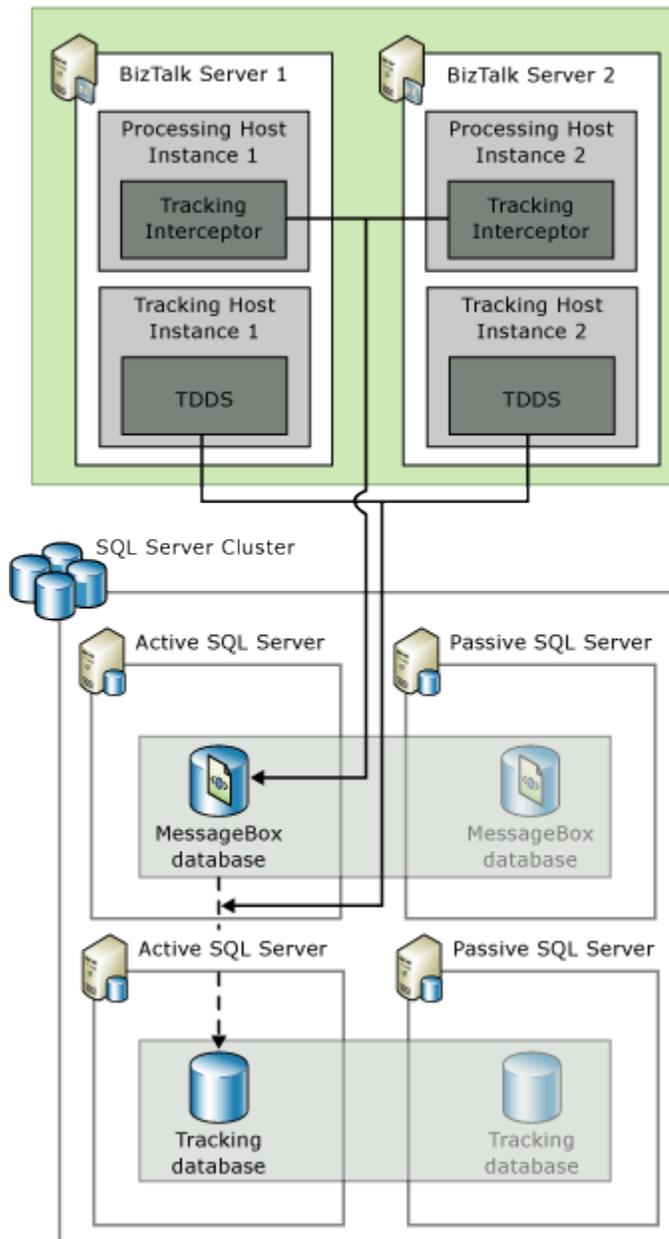
BizTalk Server stores all the states in the MessageBox databases, and each MessageBox database contains state information for different business processes. For reliability, you must cluster all the MessageBox databases, including the master and secondary MessageBox databases.

Providing High Availability for Multiple MessageBox Databases

While adding MessageBox databases to your BizTalk Server deployment improves scalability, it does not provide high availability because each MessageBox database is unique and independent and is potentially a single point of failure for your BizTalk Server environment. To add redundancy involves configuring a server cluster for each MessageBox database. BizTalk Server distributes data across the multiple MessageBox databases, so the databases do not share data or otherwise provide redundancy without server clustering.

Providing High Availability for the BizTalk Tracking Database

Depending on the requirements of your particular deployment, you might want to enhance performance for tracking by isolating the BizTalk Tracking database onto a separate SQL Server computer and by creating a separate BizTalk Host dedicated to host tracking. The following figure shows a dedicated tracking host with two host instances and clustered databases.



If your deployment has high throughput and involves tracking lots of data for these messages, the tracking overhead could potentially consume lots of resources on the computer that is running SQL Server. If this situation occurs and a high rate of incoming messages continues, BizTalk Server reaches a point where it cannot process new messages because the resources required to track messages are greater than the resources required to run the other BizTalk Server components (such as receiving messages and persisting them to the MessageBox database). To improve performance and security, we recommend that you dedicate a host for tracking that does not contain any other items (such as receive locations, orchestrations, or pipelines) and that you disable tracking from the receiving, processing, and sending hosts. To provide high

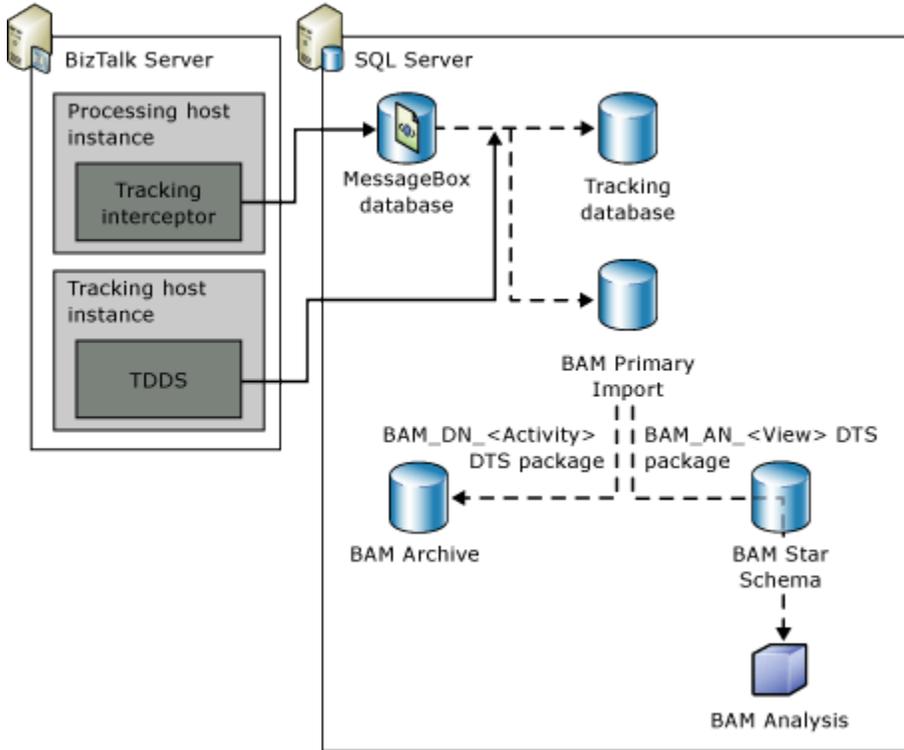
availability for the tracking host, create more than one host instance of the tracking host. For more information about creating a tracking host, see [How to Create a New Host](http://go.microsoft.com/fwlink/?LinkId=156825) (<http://go.microsoft.com/fwlink/?LinkId=156825>) in BizTalk Server Help.

For each MessageBox database, BizTalk Server uses only one tracking host instance to move messages from the MessageBox database to the BizTalk Tracking database (BizTalkDTADb). If additional computers run instances of the tracking host, BizTalk Server automatically scales out the handling of each MessageBox database to a separate instance of the tracking host. If the number of MessageBox databases is larger than the number of tracking host instances, one or more tracking host instances will service more than one MessageBox database.

To provide high availability for the BizTalk Tracking database, use Windows Clustering to configure two database computers that are running SQL Server 2008 R2 or SQL Server 2008 SP1 in an active/passive configuration.

Providing High Availability for the BAM Databases

Business Activity Monitoring (BAM) provides visibility into business processes independent of the IT implementation or across a heterogeneous IT implementation. The BAM SQL Server databases (BAM Star Schema database, BAM Primary Import database, and BAM Archive database) and the BAM Analysis database store the business activity data that is different from the operational monitoring data. The following diagram shows the BAM database infrastructure.



To make sure your BAM infrastructure is highly available, do the following:

- **Cluster the BAM Primary Import database and the BAM Analysis database.** The BAM Primary Import database is the center of the Business Activity Monitoring system. It is therefore important that you make this database highly available by using Windows Clustering and that you follow the next two recommendations to prevent this database from filling up. The BAM Analysis database is an Analysis Services database that stores the data that business analysts use to build activity aggregations and OLAP cubes, and therefore any downtime of this database affects their productivity. While you do not have to cluster the BAM Archive database, we recommend that you monitor the event log for errors when the SQL Server Integration Services (SSIS) packages run to make sure the data has been transferred successfully, and to monitor the size of the database so that you can replace it before it fills up.
- **Define an online window.** To allow for better performance and avoid downtime, BAM partitions the data in the BAM Primary Import database into tables based on the timestamp when the activity was completed. BAM achieves this by regularly swapping the completed table with another empty table of identical format. After BAM does this, the additional completed activities go into the new partition (table), while BAM keeps the old partitions for the time defined in the online window. You must define an online window to make sure the number of partitions in the BAM Primary Import database does not grow too large. For more information about scheduling online windows, see [Archiving Primary Import Database Data](http://go.microsoft.com/fwlink/?LinkId=156826) (<http://go.microsoft.com/fwlink/?LinkId=156826>) in BizTalk Server Help.
- **Schedule SSIS packages to run periodically.** Defining an online window makes sure your BAM Primary Import database does not fill up with old activity partitions. You must also schedule SSIS packages to run periodically to create a new partition for the activity data and to move the data from the old partitions in the BAM Primary Import database into the BAM Archive database. For more information about scheduling SSIS packages, see [Scheduling SQL Server Integration Services Packages](http://go.microsoft.com/fwlink/?LinkId=156827) (<http://go.microsoft.com/fwlink/?LinkId=156827>) in BizTalk Server Help.
- **Carefully choose small sets of data items (checkpoints), and avoid including unnecessary data items when defining an activity.**
- **Understand the trade-offs between scheduled and real-time aggregations when you design your aggregations.** Real-time aggregations are automatically maintained by SQL Server triggers and have zero latency. They are ideal for some mission-critical low-latency scenarios, but they incur a performance cost whenever the events are being written to the BAM Primary Import database. Scheduled aggregations rely on scheduled cubing SSIS packages to update their aggregation data. Their latency is equal to or greater than the SSIS schedule interval, but overall they have a smaller performance impact on the BAM Primary Import database.
- **If you choose scheduled aggregations, make sure you schedule the cubing SSIS to run more frequently than the archiving SSIS.** This is because the archiving SSIS will not move the activity data that has been processed for scheduled aggregation to the BAM Archive database.
- **Enable the BAM Event Bus service in multiple computers to obtain failover functionality.**

Providing High Availability for the Other BizTalk Server Databases

To provide high availability for the other BizTalk Server databases, configure two computers that are running SQL Server 2008 R2 or SQL Server 2008 SP1 in a Windows cluster. These computers can run in an active/active or active/passive configuration for redundancy and can store data on a shared drive (such as a RAID 1+0 SCSI disk array) or storage area network (SAN).

See Also

[Clustering the BizTalk Server Databases](#)

High Availability for the Master Secret Server

Even if you do not use the Enterprise Single Sign-On (SSO) functionality for mapping credentials and single sign-on, SSO is a critical part of the overall Microsoft BizTalk Server 2010 infrastructure, because BizTalk Server uses SSO to help secure information for port configuration. The port configuration data is encrypted and stored in the SSO database. Each BizTalk server has an SSO service (ENTSSO.exe) that is used for encrypting and decrypting the port configuration data.

When an SSO service starts up, it retrieves the encryption key from the master secret server. This encryption key is called the master secret. The master secret server is another SSO service that has an additional subservice that maintains and distributes the master secret. After a master secret is retrieved, the SSO service caches it. Every 60 seconds, the SSO service synchronizes the master secret with the master secret server.

If the master secret server fails, and the SSO service detects the failure in one of its refresh intervals, the SSO service and all run-time operations that were running before the server failed, including decryption of credentials, continue successfully. However, you cannot encrypt new credentials or port configuration data. Therefore, the BizTalk Server environment has a dependency on the availability of the master secret server.

Making the Master Secret Server Available

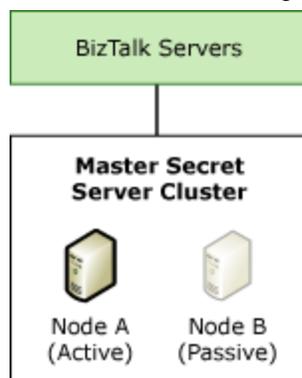
For availability of the SSO system, and therefore of the BizTalk Server environment, it is critical that you back up the master secret as soon as it is generated. If you lose it, you lose the data that the SSO system encrypted by using that master secret. For more information about backing up the master secret, see [How to Back Up the Master Secret](#) (<http://go.microsoft.com/fwlink/?LinkID=151934>) in BizTalk Server 2010 Help.

You can make the master secret server available in two ways:

- **Available, but not highly available.** You can create a Microsoft System Center Operations Manager event to notify you when the master secret server becomes unavailable, and you can then manually promote another SSO server to master secret server and restore the master secret on this server.

Although this configuration is not highly available, it can be satisfactory for most scenarios and it is consistent with scaling out the receiving, sending, and processing hosts.

- **Highly available.** To provide redundancy for the master secret server, use Windows Clustering on a separate master secret server cluster, or configure the master secret server on an existing database cluster. The services provided by the master secret server do not consume many resources, and typically do not affect database functionality or performance when installed on a database cluster. The following figure shows how you can make the master secret server highly available.



While this configuration is highly available, it requires additional hardware resources. For more information about high-availability installation options for SSO, see [High-Availability SSO Installation Options](http://go.microsoft.com/fwlink/?LinkId=156838) (<http://go.microsoft.com/fwlink/?LinkId=156838>) in BizTalk Server 2010 Help.

This section includes detailed information about configuring the SSO master secret server as a highly available cluster resource on a Windows Server cluster.

 **Note**

To reduce the hardware resources for a highly available solution, you can add the master secret server as a cluster resource in your SQL Server cluster. Note that you do not need to purchase additional BizTalk Server licenses to install the SSO service on the computer running the SQL Server.

In This Section

- [Clustering the Master Secret Server](#)
- [Designating a New Master Secret Server Manually](#)

See Also

[Planning for High Availability](#)

[High Availability for BizTalk Hosts](#)

[High Availability for Databases](#)

Clustering the Master Secret Server

The BizTalk Server 2010 application service maintains a hard-coded dependency upon the Enterprise Single Sign-On (SSO) service that is installed with BizTalk Server. The SSO service must be able to communicate with the master secret server to start. We recommend that you cluster the SSO service on the master secret server to provide fault tolerance for the master secret server. For more information, see [High-Availability SSO Installation Options](#) (<http://go.microsoft.com/fwlink/?LinkId=156838>) in BizTalk Server 2010 Help.

Preparing for Clustering the Master Secret Server

Deciding Whether to Use a Dedicated Cluster or the SQL Server Cluster

Clustering hardware is expensive. To reduce the hardware resources for a highly available solution, you can add the master secret server as a cluster resource in your SQL Server database cluster. If you use the SQL Server database cluster, we recommend that you do not put the master secret server on the same active node as the MessageBox database. The MessageBox database is usually busier than other databases. Even though the master secret server doesn't consume many hardware resources, it is better to move it to a different active cluster node from the active MessageBox database node.

Clustering the SSO Database

The master secret server depends on the SSO database. To build a high-availability SSO, you must cluster the SSO database. For more information about clustering BizTalk databases, see [Clustering the BizTalk Server Databases](#).

Creating Domain Groups and Accounts

You need to configure the following domain groups and accounts before clustering the master secret server:

- Create domain groups with the names SSO Administrators and SSO Affiliate Administrators. To create a clustered instance of the Enterprise SSO service, you must create the SSO Administrators and SSO Affiliate Administrators groups as domain groups.

- Create or designate a domain account that is a member of the SSO Administrators domain group. The Enterprise SSO service on each node will be configured to log on as this domain account. This account must have the "Log on as a service" right on each node in the cluster. This account must also be granted Full Control access to the cluster.

Clustering the Master Secret Server

Here are the basic steps for clustering the master secret server:

1. Install and configure Enterprise SSO on the cluster nodes.
2. Create the clustered Enterprise SSO resource and the dependent resources.
3. Restore the master secret on the second cluster node. If you move the master secret server to the cluster, you must restore the master secret on the first cluster node as well.
4. Bring the cluster group that contains the SSO service, online.
5. Update the master secret name in the Management database.

For detailed steps on clustering the master secret server, see [How to Cluster the Master Secret Server](http://go.microsoft.com/fwlink/?LinkId=156839) (<http://go.microsoft.com/fwlink/?LinkId=156839>) in BizTalk Server Help.

See Also

[Designating a New Master Secret Server Manually](#)

Designating a New Master Secret Server Manually

Cluster hardware can be expensive. If hardware cost is a concern, you can consider manually designating another Enterprise Single Sign-On (SSO) server to be the master secret server during failure scenarios. Using this option, any other SSO server in the SSO group can be promoted to the master secret server. When the master is down, you can manually promote one of the SSO servers to be the master secret server. The biggest disadvantage of this technique is that you cannot edit the existing deployments, restart the existing BizTalk Server services, or deploy new BizTalk applications until you promote a new master secret server.

To make the process seamless, you will need to implement some monitoring mechanism so you will discover the failure as soon as possible. You can also automate the promotion process by using a monitoring application such as System Center Operations Manager.

For more information about manually moving the master secret server, see [How to Move the Master Secret Server](http://go.microsoft.com/fwlink/?LinkId=156841) (<http://go.microsoft.com/fwlink/?LinkId=156841>) in BizTalk Server Help.

See Also

[Clustering the Master Secret Server](#)

Disaster Recovery

Disaster recovery procedures improve system availability by employing steps to resume operation of a failed system. Disaster recovery differs from fault tolerance in that disaster recovery typically requires manual intervention to restore the failed system whereas a fault-tolerant design can continue to operate without manual intervention if the system encounters a failure condition.

In This Section

- [Planning for Disaster Recovery](#)
- [Configuring for Disaster Recovery](#)
- [Recovering from a Disaster](#)
- [Advanced Information About Backup and Restore](#)

Planning for Disaster Recovery

This topic provides guidance to application teams for disaster recovery requirements and procedures for BizTalk Server 2010. Microsoft BizTalk Server 2010 stores configuration and processing information in SQL Server. BizTalk Server 2010 high availability and disaster recovery is achieved through the high availability and disaster recovery capabilities of SQL Server.

A BizTalk group is defined by a set of databases hosted in SQL Server. The set of databases hosted in SQL Server can be made highly available through the use of a Windows Server cluster. In a BizTalk environment, the computers running BizTalk Server provide the “run-time environment” and the databases on the computers running SQL Server provide the persistent store for the environment. Therefore, BizTalk Server 2010 backup, restore, and disaster recovery procedures are heavily focused on SQL Server.

Purpose

This topic consolidates BizTalk Server disaster recovery information from the core documentation, various Microsoft Web sites, and information from the BizTalk Server product team to define disaster recovery procedures for BizTalk Server environments.

Application teams should thoroughly test backup, restore, and disaster recovery procedures. You should also ensure the procedures are tailored to meet application requirements before entering production.

Scope

This section focuses on disaster recovery procedures for BizTalk Server 2010 and related procedures for SQL Server. This guide builds on related documentation about how to back up and restore BizTalk Server.

For more information about backup and restore, see this documentation and see [Backing Up and Restoring BizTalk Server](http://go.microsoft.com/fwlink/?LinkID=154071) (http://go.microsoft.com/fwlink/?LinkID=154071) in BizTalk Server 2010 Help. Each application team must augment the information in this topic with additional procedures specific to their environment, such as document computer names, drive letters, and cluster configuration, as well as disaster recovery procedures for related non-BizTalk applications that are part of the solution.

This topic does not provide detailed disaster recovery procedures for the following areas:

- Non-BizTalk applications
- Application source code
- Certificates
- Application operations

There may be additional areas that require documentation in an application's disaster recovery plan not listed above that must be addressed by each application team.

In This Section

- [Best Practices for Disaster Recovery](#)
- [What Is BizTalk Server Log Shipping?](#)

Best Practices for Disaster Recovery

For information about best practices for disaster recovery for BizTalk Server 2010, see [Best Practices for Backup and Restore](http://go.microsoft.com/fwlink/?LinkID=151391) (http://go.microsoft.com/fwlink/?LinkID=151391) in BizTalk Server 2010 Help.

Create a backup and restore plan

- Be sure your backup plan specifies:
 - The computer where backups will be stored
 - The programs that you will use to back up your system
 - The computers you want to back up
 - The schedule when backups will occur
 - The offsite location where you will archive backups

Keep a written record of all changes to your BizTalk Server system

- Be sure to write down all changes to your system, such as service packs, hotfixes, and QFEs that have been applied. This is crucial to getting your system restored as closely as possible to what existed before the hardware failure.

Implement the following measures to help prevent or minimize the effect of a disaster

- Have your software and firmware updates available.
- Have all software disks readily available.
- Have a plan to monitor servers proactively. This is very important since orchestration instances on a failed host may not be recovered by a second host for up to 10 minutes.
- Maintain hardware records.
- Maintain software records.

Implement fault tolerance in your organization at the hardware or software level

- Implementing clusters and redundant array of independent disks (RAID) helps ensure that your system can survive a hardware failure.

Archive the backup media on a regular basis in a secure location

- Create a schedule for archiving your backup media on a regular basis and keep the archives in a secure, offsite location. This ensures that you have a backup available when you need it.

Verify the integrity of your backups and that they occur without error

- Monitor all of your backup jobs and ensure that they complete without any errors.

Keep identical spare hardware available

- Having identical spare hardware available ensures that you can quickly replace defective hardware to get up and running more quickly.

Document and test your recovery procedures

- Disaster recovery testing should be conducted before ever running your system in production. Having plans in place and performing prerelease testing will ensure that your IT staff can recover your BizTalk Server systems.

Train your IT staff on disaster recovery procedures

- Ensure that your IT staff is prepared to recover the system should the need arise.

Practice restoring from a backup in a test environment

- Practice restoring your BizTalk Server system in a test environment to ensure that you can restore it to your production environment if a failure occurs.

See Also

[Disaster Recovery](#)

What Is BizTalk Server Log Shipping?

BizTalk Server 2010 disaster recovery procedures are built around BizTalk log shipping. BizTalk log shipping simplifies database restoration in the event of a disaster by continuously applying transaction log updates to the disaster recovery site databases.

While BizTalk log shipping is based on similar principles as SQL Server log shipping, SQL Server log shipping is not supported for the BizTalk Server databases backed up as part of the Backup BizTalk Server SQL Agent Job.

How Does BizTalk Log Shipping Work?

BizTalk log shipping functions in a manner similar to SQL Server log shipping. The production BizTalk Server group is configured to back up the BizTalk Server databases to a UNC location. By default, the Backup BizTalk SQL Agent job performs a full backup every hour and a log backup every 15 minutes. The Backup BizTalk Server job implements logic to automatically start a full backup if a backup failure is detected.

When the disaster recovery SQL Server instances are configured for BizTalk log shipping, the backup files created by the Backup BizTalk Server SQL Agent job are restored at the disaster recovery site every 15 minutes by default. The backup files are copied over the network by a SQL RESTORE command. Full backup files are copied only in the following situations:

- When BizTalk log shipping is first configured
- When a new database is added to the Backup BizTalk Server job.
- When a RESTORE failure occurs at the disaster recovery site

Each SQL Server instance at the disaster recovery site is configured individually as part of BizTalk log shipping to restore databases hosted on a production SQL Server database instance. When a SQL Server instance is configured for BizTalk Server log shipping and the **BTS Log Shipping Restore Databases** job is enabled, the **BTS Log Shipping Restore Databases** job will connect to the BizTalk Management database on the production BizTalk Server group.

As described above, when the destination server is first configured the full database backup is restored to the destination server. Most of the time only the logs are restored when the **BTS Log Shipping Restore Databases** job runs.

When viewing the disaster recovery SQL Server instances with SQL Server Management Studio, the databases will be displayed in a “Loading” state. This is because the last log in a backup set is never restored automatically. Once a new log is available, BizTalk Server log shipping restores the next to last log. When a disaster recovery event occurs and the disaster recovery site must be brought online, the last log is restored using the STOPATMARK command to recover the databases and the databases will no longer be displayed as being in a “Loading” state.

Configuring for Disaster Recovery

The BizTalk Server 2010 Log Shipping feature extends the existing Backup BizTalk Server job. BizTalk Server 2010 Log Shipping eliminates the need to manually restore a series of backup sets produced by the backup job, and reduces downtime in the event of a system failure. BizTalk Server 2010 Log Shipping is a critical component for BizTalk disaster recovery procedures.



Note

Each application team must have a documented backup and restore plan for disaster recovery that complements the concepts provided in this topic. The overall plan should address the entire system, including applications and the components of the operating system.

Performing a disaster recovery operation is very similar to manually performing a restore of a BizTalk group to a new set of SQL Server database instances. The primary difference is that BizTalk Server 2010 log shipping applies logs continuously at the disaster recovery site, saving many manual steps. Therefore, only the last set of logs must be restored manually when BizTalk Server 2010 log shipping is implemented. Otherwise, the last full backup followed by all log backups since the last full backup would have to be manually restored. BizTalk Server 2010 log shipping reduces the effort for this manual process, speeding restoration of the disaster recovery site.

This section covers recommendations on production configuration to facilitate the disaster recovery process.

In This Section

- [Prepare the Disaster Recovery Site](#)
- [Log Shipping User Accounts and Roles](#)
- [Configuring BizTalk Server Log Shipping](#)

See Also

[Disaster Recovery](#)

Prepare the Disaster Recovery Site

BizTalk Server 2010 log shipping has two supported scenarios. One is where log shipping for all databases on all production instances of SQL Server is applied to a single disaster recovery instance of SQL Server. The other scenario is where log shipping for the databases for each production instance of SQL Server is applied to a corresponding instance of SQL Server at the disaster recovery site. Note that it is fully supported to have the same number of SQL Server database instances in the disaster recovery site as there is in production but on fewer physical servers. This section provides guidance on these preparations.

In This Section

- [Preparing the Disaster Recovery SQL Servers](#)
- [Backing Up the BAM Analysis and Tracking Analysis Server Databases](#)
- [Preparing the Disaster Recovery BizTalk Servers](#)
- [Preparing Applications for Disaster Recovery](#)

- [How to Restore the Master Secret Server](#)

Preparing the Disaster Recovery SQL Servers

Create a set of SQL Server database instances in the disaster recovery site. To ensure that the disaster recovery SQL Server database instances can provide the same level of performance as the production SQL Server database instances, the disaster recovery SQL Server database instances should be configured with similar hardware and number of physical computers running SQL Server. In this scenario, BizTalk Server 2010 log shipping will be configured for each production SQL Server database instance to apply to a corresponding SQL Server database instance at the disaster recovery site.

A key BizTalk Server 2010 log shipping requirement is that the drive letter(s) on the production site where the database files are stored match the drive letter(s) at the disaster recovery site where the database files are restored. So if the SQL Server database file group is located on G:\data in production, there must be a G:\data directory on the destination (DR) server, or the restore will fail.

BizTalk Server 2010 log shipping does not support the **RESTORE WITH MOVE** SQL Server command. Because of this, we recommend that database instance names at the disaster recovery site match the database instance names in production (by default, the instance name is part of the file path). Another option is to simply create a directory on the corresponding drive letter in the disaster recovery computer running SQL Server so that the **RESTORE** operation can create the file in the same directory structure as is used in production.

Create SQL Server security logins for the disaster recovery site that correspond to the production site so that in the event that a failover to the disaster recovery site is required, all required security logins are present on the destination system.

Once installation of the disaster recovery SQL Server instances is completed, perform a full backup of master and msdb databases so that a clean system can be restored in the event that a switch to the disaster recovery site fails.

Backing Up the BAM Analysis and Tracking Analysis Server Databases

For information about backing up the BAM Analysis and Tracking Analysis Server databases, see [How to Back Up the BAM Analysis and Tracking Analysis Server Databases](#) (<http://go.microsoft.com/fwlink/?LinkId=151392>).

Preparing the Disaster Recovery BizTalk Servers

Install BizTalk Server run-time servers at the disaster recovery site following the recommendations in [BizTalk Server 2010 Installation and Upgrade Guides](http://go.microsoft.com/fwlink/?LinkID=194815) (<http://go.microsoft.com/fwlink/?LinkID=194815>). Configure these BizTalk Server run-time servers using the BizTalk Configuration Wizard to join them to the production BizTalk group. When configuring the BizTalk Server run-time servers at the disaster recovery site (including the disaster recovery Enterprise Single Sign-On Master Secret server) make sure to:

- Select **No** to the question “Is this the master secret server?”
- Select **Join** to the question “Do you want to create or join a BizTalk Server group?”

After configuring the BizTalk Server run-time disaster recovery servers, create BizTalk host instances on the disaster recovery site that correspond to the production site host instances, but do not start these host instances. For example, if the production site has three Hosts **Send**, **Receive**, and **Orchestration** with instances on server1, server2, and server3, create three corresponding host instances on DRserver1, DRserver2, DRserver3.



Note

All BizTalk Server-related Windows services such as the BizTalk Host Instance and Rules Engine Service at the disaster recovery site should be set to “disabled” in the Services Manager to prevent the disaster recovery site from performing any processing.

You can install and configure a different number of BizTalk Server run-time servers in the disaster recovery site than are running in production. Use caution when taking this approach however, so as to avoid overloading the servers in the disaster recovery site if a disaster recovery event occurs. Once the BizTalk group as represented by the set of databases is fully restored, and all required system changes are performed for the BizTalk group, scripts can be run to join the BizTalk Server run-time servers to the BizTalk group.

Preparing Applications for Disaster Recovery

BizTalk applications (binaries and configuration artifacts such as receive locations and send ports) are deployed to the production BizTalk group when the group is restored at the disaster recovery site. This configuration may have to be altered depending on whether there are configuration locations such as receive locations and send ports that are production-specific.

To speed the restoration of applications at the disaster recovery site, a BizTalk Server .msi file that installs the binaries on the BizTalk Server run-time servers must be kept up-to-date on the disaster recovery BizTalk Server run-time servers. When the production BizTalk group is restored at the disaster recovery site, a disaster recovery-specific application configuration .msi file may need to be installed in order to configure the application for disaster recovery-specific artifacts, such as receive locations and send ports, as necessary.

See Also

[Deploying an Application](#)

How to Restore the Master Secret Server

For information about restoring the master secret server, see [How to Restore the Master Secret](#) (<http://go.microsoft.com/fwlink/?LinkId=151394>).



Note

In order to restore the SSO master secret, you must have previously created a backup of the SSO master secret by following the steps at [How to Back Up the Master Secret](#) (<http://go.microsoft.com/fwlink/?LinkId=151395>).

Log Shipping User Accounts and Roles

BizTalk Server 2010 log shipping is driven by a SQL Server Agent job to automate the process of restoring backups and logs. Incorrect permissions can cause restore operations performed by BizTalk Server 2010 log shipping to fail. The user account configured to restore databases must have access to the production database instance that hosts the BizTalk Management database. In most cases this means that the service account for the SQL Server Agent job driving the BizTalk Server 2010 log shipping job on the disaster recovery SQL Server instance requires a login and permissions on the production database instance that hosts the BizTalk Server management database. This assumes that the SQL Server Agent service account is configured as the job owner.

BizTalk Server 2010 includes a SQL Server role named `BTS_BACKUP_USERS` so that the user account configured to restore databases does not require SQL Server System Administrators permission.

When configuring the user account that will perform database restore operations as part of BizTalk Server 2010 log shipping, please verify the following:

- Configure the SQL Server Agent service to run under a domain account with a mapped user configured in SQL Server Management Studio (in **Security, Logins**) on each SQL Server instance that hosts BizTalk Server databases restored by the BizTalk Server log shipping jobs.
- Configure a SQL Server login account for this user, and assign this user to the BizTalk `BTS_BACKUP_USERS` SQL role on each server.
- Assign this user to the SQL Server System Administrators role on the computer running SQL Server that houses the BizTalk Server management database.
- The account that performs backup and restore operations must have Read and Write access to the UNC share where backup files are created.

Configuring BizTalk Server Log Shipping

You use the Backup BizTalk Server job to back up all of the databases in your BizTalk Server source system, except for some databases used by Business Activity Monitoring (BAM). The source system is the server or group of servers that contain live data. Because some of the BAM databases have different backup and restore requirements, these databases are backed up and restored using other methods.

Backing up the BizTalk Server databases and restoring them involves the following steps:

- 1. Configuring the Backup BizTalk Server job**

Before you can back up the BizTalk Server databases, you must first configure the Backup BizTalk Server job on the source system, which directs backups to be automatically written to a folder where they can then be used to restore the databases on the destination system.

The destination system is the server or group of servers that will be used to restore the database backups produced by the source system. For more information about this step, see [How to Configure a Backup BizTalk Server Job](#).

- 2. Configuring the destination system for log shipping**

You must also configure the destination system for log shipping, which provides standby server capabilities and reduces downtime in the event of a system failure. For more information about this step, see [How to Configure the Destination System](#).

- 3. Restoring the databases**

When a hardware failure occurs, you can restore your databases by using the backups and logs sent to your destination system. For more information about this step, see [How to Restore Databases in the Backup BizTalk Server Job](#).

BizTalk Server Databases

The following tables describe the databases used by BizTalk Server and identify which methods are used to back up the databases.

Databases Backed Up by the Backup BizTalk Server Job

The following table lists the databases that are backed up and restored as a part of the Backup BizTalk Server job. You can modify the Backup BizTalk Server job to back up custom databases by adding them to the adm_OtherBackupDatabases table. For more information, see [How to Back Up Custom Databases](#) (<http://go.microsoft.com/fwlink/?LinkID=151569>).

Database	Default database name	Description
BAM Primary Import database	BAMPrimaryImport	This is the database where the Business Activity Monitoring (BAM) collects raw tracking data.

Database	Default database name	Description
BAM Notification Services Application database	BAMAlertsApplication	This database contains alert information for BAM notifications. For example, when you create an alert using the BAM portal, entries are inserted in the database specifying the conditions and events to which the alert pertains, as well as other supporting data items for the alert.
BAM Notification Services Instance database	BAMAlertsNSMain	This database contains instance information specifying how the notification services connect to the system that BAM is monitoring.
BizTalk Tracking database	BizTalkDTADb	This database stores health monitoring data tracked by the BizTalk Server tracking engine.
BizTalk Management database	BizTalkMgmtDb	This database is the central meta-information store for all instances of BizTalk Server.
BizTalk MessageBox database	BizTalkMsgBoxDb	This database is used by the BizTalk Server engine for routing, queuing, instance management, and a variety of other tasks.
Rule Engine database	BizTalkRuleEngineDb	This database is a repository for: <ul style="list-style-type: none"> • Policies, which are sets of related rules. • Vocabularies, which are collections of user-friendly, domain-specific names for data references in rules.
SSO database	SSODB	This Enterprise Single Sign-On database more securely

Database	Default database name	Description
		stores the configuration information for receive locations.

Databases Backed Up When Backing Up the BAM Analysis and Tracking Analysis Server Databases

The following table lists the databases that are backed up using the procedures in **Backing Up the BAM Analysis and Tracking Analysis Server Databases**:

Database	Default database name	Description
BAM Star Schema	BAMStarSchema	This database contains the staging table, and the measure and dimension tables.
BAM Analysis	BAMAnalysis	This database contains BAM OLAP cubes for both online and offline analysis.
BAM Archive	BAMArchive	This database archives old business activity data. Create a BAM Archive database to minimize the accumulation of business activity data in the BAM Primary Import database.
Tracking Analysis Server	BizTalkAnalysisDb	This database stores health monitoring online analytical processing (OLAP) cubes.

This section of the operations guide describes additional steps that you should follow to configure BizTalk Server log shipping.

In This Section

- [Configuring the Source System](#)
- [How to Configure the Destination System](#)
- [Configuring BizTalk Server Log Shipping for Additional Databases](#)
- [Monitoring BizTalk Server Log Shipping](#)

Configuring the Source System

For the purposes of BizTalk Server 2010 log shipping, it does not matter if the source system is located on a single SQL Server instance or if it is distributed among multiple instances hosted in a Windows Server cluster. There are no additional considerations other than those required to successfully run the Backup BizTalk Server job. To configure this job, see [How to Configure the Backup BizTalk Server Job](http://go.microsoft.com/fwlink/?LinkID=154072) (http://go.microsoft.com/fwlink/?LinkID=154072) in BizTalk Server 2010 Help. After you have configured the Backup BizTalk Server Job, proceed to the topic [How to Configure the Destination System](#).

See Also

[Configuring BizTalk Server Log Shipping](#)

How to Configure the Destination System

For information about configuring the destination system for log shipping, see [How to Configure the Destination System for Log Shipping](http://go.microsoft.com/fwlink/?LinkID=151402) (http://go.microsoft.com/fwlink/?LinkID=151402) in BizTalk Server 2010 Help.

See Also

[What Is BizTalk Server Log Shipping?](#)

[How to Configure a Backup BizTalk Server Job](#)

[How to Schedule a Backup BizTalk Server Job](#)

Configuring BizTalk Server Log Shipping for Additional Databases

In BizTalk Server, jobs added to the Backup BizTalk Server job are automatically added to BizTalk Server log shipping. You do not have to take additional steps to configure log shipping for new databases added to the Backup BizTalk Server job. However, be sure to add custom databases as appropriate under the <OtherDatabases> section of the SampleUpdateInfo.xml file as described in step 22 of [How to Configure the Destination System for Log Shipping](http://go.microsoft.com/fwlink/?LinkID=151402) (http://go.microsoft.com/fwlink/?LinkID=151402) in BizTalk Server 2010 Help.

See Also

[Configuring BizTalk Server Log Shipping](#)

Monitoring BizTalk Server Log Shipping

To determine the last successful backup set of BizTalk Server databases and logs that have been restored, review the contents of the Master.dbo.bts_LogShippingHistory table on the destination SQL Server instance(s). This table is populated by the BizTalk Server Log Shipping Get Backup History job and is updated by the restore job. When a backup is successfully restored, the Restored column is set to a value of 1 and the RestoredDateTime is set to the current date and time.

When all of the databases restored to the server from a particular backup set listed in the Master.dbo.bts_LogShippingHistory table have been successfully restored, the backup set ID is written to the Master.dbo.bts_LogShippingLastRestoreSet table. This table stores the last set restored and is useful for determining what backup set of logs need to be restored to bring the destination BizTalk group online after the occurrence of a disaster recovery event.

See Also

[Configuring BizTalk Server Log Shipping](#)

Recovering from a Disaster

This section details the steps to transfer production operations to the disaster recovery site. For the purposes of this section, the online production system is referred to as the “source” system. The disaster recovery site is referred to as the “destination” system. This section provides an overview of the disaster recovery process.

In This Section

- [Restoring the BizTalk Group](#)
- [Recovering the Runtime Computers](#)
- [Recovering Additional Applications](#)
- [Troubleshooting Log Shipping](#)

Restoring the BizTalk Group

The BizTalk group is represented by the set of SQL Server and SQL Server Analysis Services databases, SSIS packages, and SQL Agent Jobs. This section describes the process for restoring the BizTalk group.

In the event that a switchover to the destination system (disaster recovery site) is required, the following steps must be completed:

1. Restore SQL Server and Analysis Services databases.
2. Restore BizTalk Server 2010 runtime servers and applications.

Upon completion of these steps, the BizTalk group has been established at the disaster recovery site, the BizTalk Server runtime servers can be configured, and the applications can be deployed into the BizTalk group. The topics in this section cover the details of this process.

In This Section

- [Stopping Application Processing on the Source System](#)
- [How to Restore Databases in the Backup BizTalk Server Job](#)
- [Restoring Databases Not Included in the Backup BizTalk Server Job](#)

Stopping Application Processing on the Source System

Application processing should be stopped when the source BizTalk Server runtime servers are still able to participate in document processing using the existing database servers. In this scenario, processing activity must be stopped so that a consistent restore operation can be completed.

To stop application processing on the source system, ensure that no connections are open between the production BizTalk Server runtime computers and the SQL Server computers that house the BizTalk Server databases. Follow these steps to stop application processing on the production BizTalk Server runtime computers:

1. Disable all receive locations on the BizTalk Server computers in the BizTalk group. Make a note of all receive locations that are disabled so that these receive locations can be re-enabled later. This will stop BizTalk Server from processing incoming messages.
2. Stop all host instances from running on the BizTalk Server computers in the group. This can be done from the BizTalk Server Administration console. Make a note of all the host instances that were stopped so that these host instances can be restarted later.
3. Stop all SQL Server Agent jobs related to BizTalk Server on the SQL Server computers that house BizTalk Server databases.
4. If BAM is in use, disable any BAM cube update and data maintenance SSIS packages. This can be done by using SQL Server Management Studio.
5. Stop Analysis Services on the SQL Server computers that house BizTalk Server databases. This can be done by stopping all instances of MSSQLServerOLAPService on the SQL Server computers where Analysis Services is installed.
6. Stop any other BizTalk Server services in Services Manager that may be running on the BizTalk Server computers in the group, for example, the Enterprise Single Sign-On Service and the Rule Engine Update Service. Make a note of the services that are stopped so that they can be restarted later.

7. Close all applications that connect to the SQL Server computers that house BizTalk Server databases. This includes instances of the BizTalk Server Administration console, Visual Studio 2010 , and any other installed BizTalk applications.
8. Verify that there is no database activity generated by BizTalk Server. Use SQL Server Management Studio to see what processes are connected to the SQL Server computers that house BizTalk Server databases. This can be done by expanding **Management** and double-clicking **Activity Monitor** in SQL Server Management Studio. Then click to select **Process Info**. Alternatively use the system stored procedures **sp_who** or **sp_who2** to identify any open connections to the SQL Server computers that house BizTalk Server databases. If there are any processes connected, locate them and terminate them normally; or as a last resort, right-click each process in the **Process Info** pane in SQL Server Management Studio and click **Kill process** to terminate the connection.
9. Additional database processing may be occurring in application databases. If these databases will be restored, ensure that all processing is stopped.

See Also

[Restoring the BizTalk Group](#)

How to Restore Databases in the Backup BizTalk Server Job

This section covers the steps to bring online the databases in the BizTalk group that are backed up by the Backup BizTalk Server job. By default, all databases are backed up by using the Backup BizTalk Server job except for the BAM databases. See [Restoring Analysis Services and Supporting Databases](#) for more information about backup and restore of the BAM databases. You must restore all databases to the same mark to ensure a consistent transactional state among the databases. For more information, see [Marked Transactions, Full Backups, and Log Backups](#) (<http://go.microsoft.com/fwlink/?LinkId=151565>).

For more information and instructions about restoring databases for BizTalk Server, see [How to Restore Your Databases](#) (<http://go.microsoft.com/fwlink/?LinkId=151406>).

See Also

[Restoring Databases Not Included in the Backup BizTalk Server Job](#)

Restoring Databases Not Included in the Backup BizTalk Server Job

This section describes how to restore databases that are part of the overall BizTalk solution but are not backed up by the Backup BizTalk Server job. All databases that are part of a BizTalk solution will be backed up by using the Backup BizTalk Server job except for the following:

- SQL Server Analysis Services databases
- BAM databases when BAM is enabled and configured using BM.exe

This section also describes how to update database references after restoring the databases listed above and includes information about resolving incomplete BAM activity instances.

In This Section

- [Restoring Analysis Services and Supporting Databases](#)
- [Updating Database References](#)
- [How to Resolve Incomplete BAM Activity Instances](#)

Restoring Analysis Services and Supporting Databases

There are two SQL Server Analysis Services databases that must be restored in a disaster recovery scenario:

- BAM Analysis (BAMAnalysis)
- Tracking Analysis Server (BizTalkAnalysisdb)

The BAM SQL Server databases may be backed up as part of the Backup BizTalk Server job if BAM is enabled but not configured.



Note

The BAM Primary Import database is always backed up as part of the Backup BizTalk Server job because it participates in DTC transactions.

The following BAM databases will be part of the Backup BizTalk Server job if BAM is enabled but not configured:

- BAMStarSchema
- BAMArchive

Follow the steps in [How to Restore Databases in the Backup BizTalk Server Job](#) to restore these databases.

Otherwise, if BAM is enabled and is also configured with BM.exe, the correct set of BAM databases must be restored together as a set. To ensure that a complete set of archived data

is recovered, the BAM Archive database is backed up after the partition is copied into the BAM Archive, but before the partition is deleted from the BAM Primary Import database. This is performed by modifying the data maintenance SSIS package for each activity by inserting a step to back up the BAM Archive database before the last step "End Archiving."

The restore procedure for the BAM databases is: If the BAM Primary Import database is restored with the last backup date of x, restore the copies of the BAM Archive and BAM Star Schema databases that correspond to the latest date when the data maintenance SSIS package was run before the date of x.

After the correct set of BAM databases is identified, restore the SQL Server and SQL Server Analysis Services databases using standard procedures as documented in the SQL Server Books Online for restoring SQL Server databases and SQL Server Analysis Services databases.

See Also

[Backing Up the BAM Analysis and Tracking Analysis Server Databases](#)

Updating Database References

This section describes how to update references to various SQL Server Analysis Services databases and BAM databases after the databases have been restored.

In This Section

- [Update References to the BAM Analysis Server and Star Schema Database Names](#)
- [Update References to the Tracking Analysis Server Database](#)
- [Update References to the BAM Archive Database Name](#)
- [Update References to the BAM Primary Import Database](#)
- [Update References to the BAM Notification Services Databases](#)

Update References to the BAM Analysis Server and Star Schema Database Names

If you backed up your BAM Analysis database, in the event of a system or data failure you can restore that backup to a different computer and you can rename the backup.

To restore the BAM Analysis and Star Schema databases, perform the steps in [How to Restore Databases in the Backup BizTalk Server Job](#). In addition, you must update the BAM SQL Server Integration Services (SSIS) packages with the new server name and database name.

How to Update References to BAM Analysis Server and Star Schema Database Names

For instructions on how to update references to BAM Analysis server databases, see [Updating References to the new BAM Analysis Database](#). For instructions on how to update references to the BAM Star Schema databases, see [Updating References to the New BAM Star Schema Database](#).

Update References to the Tracking Analysis Server Database

The Tracking Analysis Server database is an optional and contains the online analytical processing (OLAP) cubes. These OLAP cubes are aggregations of data contained in the BizTalk Tracking database.

To restore the Tracking Analysis Server database, use SQL Server Analysis Manager to process the MessageMetrics and ServiceMetrics cubes. For instructions, see [Managing Backing Up and Restoring \(Analysis Services\)](#) (<http://go.microsoft.com/fwlink/?LinkId=130939>) in SQL Server Books Online.

To restore the Tracking Analysis Server database to an alternate computer, you must also update references to the database name in the BizTalk Management database by using the following procedure.

Prerequisites

You must be logged on as a member of the BizTalk Server Administrators group to perform this procedure.

► To update references to the Tracking Analysis Server database name

1. On the destination system, click **Start**, click **Programs**, click **Microsoft SQL Server 2008**, and then click **SQL Server Management Studio**.
2. In the **Connect to Server** dialog box, select the **Server type** as **Database Engine**, provide a server name, provide the credentials to connect to the server, and then click **Connect**.
3. Open the appropriate server by clicking it, double-clicking **Databases**, double-clicking **BizTalkMgmtDb**, and then clicking **Tables**.
4. In the details pane, right-click **adm_Group**, and then point to **Open Table**.
5. Modify the columns corresponding to the original database to reference the appropriate values for the new database.



<DBType> DBServerName and <DBType> DBName indicate the location of the database, where <DBType> corresponds to the type of the database, for example, TrackingAnalysis.

6. Close the table to save the new values.

See Also

[Updating Database References](#)

Update References to the BAM Archive Database Name

If you backed up your BAM Archive databases, in the event of a system or data failure you can restore that backup to a different computer, and you can rename the backup.

To restore the BAM Archive databases, perform the steps in [How to Restore Databases in the Backup BizTalk Server Job](#). In addition, you must update the BAM SSIS packages with the new server name and database name.

How to Update References to BAM Archive Database

For instructions on how to update references to BAM Archive database, see [Updating References to the New BAM Archive Database](#).

See Also

[Backing Up the BAM Analysis and Tracking Analysis Server Databases](#)

Update References to the BAM Primary Import Database

If you backed up your BAM Primary Import database in the event of a system or data failure, you can restore that backup to a different computer and rename the backup.

The BAM Event Bus service moves event data from the MessageBox database to the BAM Primary Import database. The BAM Event Bus service includes fault tolerance logic that enables it to recover and restart from an unexpected failure without losing any data. For more information about the BAM Event Bus service, see the topic [Managing the BAM Event Bus Service](#) (<http://go.microsoft.com/fwlink/?LinkID=154194>).

To restore the BAM Primary Import database, perform the steps in [How to Restore Databases in the Backup BizTalk Server Job](#). In addition, you must update the BAM SSIS packages with the new server name and database name.

How to Update References to BAM Primary Import Database

For instructions on how to update references to BAM Primary Import database, [Updating References to the New BAM Primary Import Database](#).

See Also

[Backing Up the BAM Analysis and Tracking Analysis Server Databases](#)

Update References to the BAM Notification Services Databases

For information about updating references to the BAM Notification Services databases, see [Updating References to the New BAM Notification Services Databases](#).

See Also

[Backing Up the BAM Analysis and Tracking Analysis Server Databases](#)

How to Resolve Incomplete BAM Activity Instances

For information about resolving incomplete BAM activity instances, see [How to Resolve Incomplete Activity Instances](#) (<http://go.microsoft.com/fwlink/?LinkId=151475>).

See Also

[Backing Up the BAM Analysis and Tracking Analysis Server Databases](#)

Recovering the Runtime Computers

After the steps in the topic [Restoring the BizTalk Group](#) have been completed, the BizTalk Server runtime computers can be restored.



Note

This section assumes that the steps in the topics [Preparing the Disaster Recovery BizTalk Servers](#) and [Preparing Applications for Disaster Recovery](#) have already been completed.

In This Section

- [How to Update the Runtime Computers](#)
- [How to Restore Applications and Enable Processing](#)

How to Update the Runtime Computers

The destination system BizTalk Server runtime computers are configured with the BizTalk Server Configuration Wizard as part of the production BizTalk group running in the production environment. When the production BizTalk group is restored in the disaster recovery environment, settings must be updated on each BizTalk Server runtime computer so that it points to the disaster recovery SQL Server instance(s) when it attempts to connect to the restored production BizTalk group. After the BizTalk group is restored in the destination system, use the following procedure to update the BizTalk Server runtime computers.

► To update the BizTalk Server runtime computers

1. Copy the edited SampleUpdateInfo.xml file to the \Program Files\Microsoft BizTalk Server 2010\Schema\Restore directory on every 32 bit BizTalk server or to the \Program Files (x86)\Microsoft BizTalk Server 2010\Bins32\Schema\Restore directory on every 64 bit BizTalk server in the destination system.
2. On each BizTalk server, open a command prompt. Click **Start**, click **Run**, type **cmd**, and then click **OK**.



Note

On 64-bit computers, you must open a 64-bit command prompt.

3. At the command-prompt, navigate to \Program Files\Microsoft BizTalk Server 2010\Schema\Restore (on 32 bit computers) or to \Program Files (x86)\Microsoft BizTalk Server 2010\Bins32\Schema\Restore (on 64 bit computers) and type this command:

```
cscript UpdateRegistry.vbs SampleUpdateInfo.xml
```

4. Enable and restart all BizTalk Host instances and all other BizTalk Services on the BizTalk servers in the destination system.
5. Restart WMI on each BizTalk server in the destination system. Click **Start**, click **Run**, type **services.msc**, and then click **OK**. In the **Services** MMC snap-in, right-click **Windows Management Instrumentation** and select **Restart**.
6. On each BizTalk server, open the BizTalk Server Administration console, right-click **BizTalk Group**, and then select **Remove**.

7. Right-click **BizTalk Server 2010 Administration**, select **Connect to Existing Group**, select the SQL Server database instance and database name that corresponds to the BizTalk Management database for the BizTalk group, and then click **OK**.

**Note**

After updating the BizTalk Server runtime computers, you may also need to update the **SSO Server name** as displayed in the **Group Properties** dialog box available in the BizTalk Server Administration console. To update the **SSO Server name**, launch the BizTalk Server Administration console, click to expand BizTalk Server Administration, right-click the **BizTalk Group** node and select **Properties** to display the **General** tab of the BizTalk Server Administration console. Then, in the **SSO Server name** textbox, enter the name of the Enterprise Single Sign-On server that this computer will use to access the configuration information for the adapters and click **OK**. This is the name of the SSO server used to connect to the SSO database.

8. Restart the following Windows services on each BizTalk runtime server:
 - Enterprise SSO Service
 - Rule Engine Update Service
 - BizTalk Host Instances

See Also

[Recovering the Runtime Computers](#)

How to Restore Applications and Enable Processing

Configure the applications in the BizTalk group and enable application processing after following the steps described in the topic [How to Update the Runtime Computers](#).

► To enable application configuration and restore application processing

1. Run the BizTalk Server Application Installation MSI file on each BizTalk server in the group.
2. Run the BizTalk Server Application Configuration MSI file on one server in the group to configure the application for the disaster recovery site. The names for receive locations and send ports remain the same. The BizTalk Server Application Configuration MSI file updates bindings so that these artifacts point to the correct locations in the disaster recovery environment.

**Note**

If receive locations and send ports are not affected by the loss of the production

site, it may not be necessary to reconfigure the application with disaster recovery-specific locations.

3. Restore application processing by enabling all application receive locations, send ports, and host instances.

See Also

[Recovering the Runtime Computers](#)

Recovering Additional Applications

There may be additional applications that must be restored as part of the overall application solution. Procedures for recovering additional applications must be defined by the application teams and made part of the overall disaster recovery plan.

See Also

[Recovering from a Disaster](#)

Troubleshooting Log Shipping

This section describes troubleshooting scenarios related to BizTalk Server log shipping.

In This Section

- [Gaps in the Restore Process](#)
- [Partial Backup Sets](#)
- [Corrupt Backup Files](#)
- [Cleaning the Destination Environment](#)
- [Resolving Login Issues After Restoring the Destination System](#)
- [Distributed System Problems](#)

Gaps in the Restore Process

When reviewing records in the Master.dbo.bts_LogShippingHistory table, you may observe gaps in restored sets. This can occur for several reasons. However, the stability of the destination system can be restored even when gaps have occurred. A gap must be followed by a restore of a full backup set to repair the destination environment. If a gap is not followed by a full backup set restore, the destination environment is not stable and cannot be restored in a consistent state.



Note

Full backups are only restored to initially create the database and to repair problems in the log history backup chain. As long as a problem does not occur with a restore, full backup sets are not restored, because they do not participate in the log backup chain.

See Also

[Troubleshooting Log Shipping](#)

Partial Backup Sets

When backing up the databases on the source system, problems may occur that result in a partial backup set. When this occurs, the Master.dbo.bts_LogShippingHistory table will contain a 0 in the **SetComplete** column for all records in the set.

These sets cannot be restored. As a result the log backup set chain is broken. The set must be ignored, as well as all log backup sets after it, up to the next full backup set. The restore job will automatically look for the next valid full backup set. If it does not find one, it fails and restores that set in order to repair the destination environment.

In most cases the source system will detect that a partial backup set has occurred and will automatically produce a full backup set the next time it runs if it is configured to do so.



Note

The most common cause of this problem is insufficient disk space for the file groups on the destination system.

See Also

[Troubleshooting Log Shipping](#)

Corrupt Backup Files

A backup file may become corrupt, damaged, or missing. If this occurs, at least one file cannot be restored. The restore job on the system that suffered the failure searches for the next valid full backup set. In most cases it will be necessary to force a full backup on the source system. If no such set exists, the restore job fails and each subsequent run also fails until a valid full backup set arrives. If a set does exist, it is used to repair the environment.



Note

Due to the manner in which SQL Server writes backup data to a file, it is possible that SQL Server will report successful completion even if the backup failed during the actual

writing of the data. This scenario primarily occurs when the disk being written to is not local to the computer and a network failure or interruption occurs. As a general precaution, if a network failure occurs while the backup job is running, force a full backup after the network connectivity problem is resolved.

See Also

[Troubleshooting Log Shipping](#)

Cleaning the Destination Environment

If the restore job encounters error conditions that cannot be resolved, clean the destination environment so that it can start from an empty environment. Running the stored procedure **sp_LogShippingClean** located in the master database on the destination SQL Server instance will “clean” the destination environment. This procedure drops all databases and deletes the last restored data set for the specified source.

Before running this procedure, disable the **BTS Log Shipping - Restore Databases** job (the get backup history job may continue running). For more information about disabling the **BTS Log Shipping - Restore Databases** job see [How to Restore Databases in the Backup BizTalk Server Job](#). After the **sp_LogShippingClean** procedure is run, the next time the restore job runs it will find the most recent full backup set with a valid subsequent log backup set. If this set has already been restored, the restore job clears the **Restored** column for the set and all subsequent sets and then proceeds with restoring the set.



Note

Because the job looks for the most recent full backup set after the environment has been cleaned, force a full backup on the source system after running this procedure but before running the restore job.



Note

The **sp_LogShippingClean** procedure must be repeated on all servers that are restoring databases for a given source system in order to keep the different servers in synch with each other in terms of which sets have been applied.

To run the **sp_LogShippingClean** procedure, connect to the master database on all SQL Server instances that are part of the disaster recovery site and execute the following command in the **New Query** option available in SQL Server Management Studio for each SQL Server instance:

```
sp_LogShippingClean 'SourceID'
```

where **SourceID** corresponds to the identifier configured on the production SQL Server instances.

See Also

[Troubleshooting Log Shipping](#)

Resolving Login Issues After Restoring the Destination System

Depending on how the source system was configured during deployment, "orphaned" users may need to be resolved. An orphaned database user is a user who does not have a corresponding security login in SQL Server. Create corresponding logins for these users before bringing the system online using the SQL Server SQL Management Studio (in **Security, Logins**). You can create these logins at any point, but we recommend that you create them when BizTalk Server log shipping is configured.

The logins that are created should correspond to the Windows accounts and groups that were used when BizTalk Server was configured on the source system and to any logins that were manually created and used in any BizTalk Server-created role. If these logins correspond to local Windows accounts or groups, the accounts and groups must first be created before the login can be added. If the computer name for the BizTalk server is not changed, then resolve the users associated with the logins for the local accounts and groups.

When configuring BizTalk Server log shipping, follow the steps in Microsoft Knowledge Base Article 918992 [How to transfer the logins and the passwords between instances of SQL Server 2005 and SQL Server 2008](http://go.microsoft.com/fwlink/?LinkId=157143) (<http://go.microsoft.com/fwlink/?LinkId=157143>) to create a script that will add the necessary logins to the destination server.

See Also

[Troubleshooting Log Shipping](#)

Distributed System Problems

In a distributed destination system the restore jobs are not aware of errors or problems on the other computers. For example, suppose that computer A is restoring the BizTalk Management database and the BizTalk Tracking database, and computer B is restoring the BizTalk MessageBox database. Both computers successfully restore backup sets 1 through 25. Set 26, however, has a corrupted log backup file of the BizTalk MessageBox database. Computer A restores its databases correctly but computer B fails to restore the corrupted file.

In this situation, force a full backup on the source system. Continuing the example above, assume that the problem was diagnosed and a full backup was created for set 35. Computer A then restores sets 26 to 34, because it is not aware of the problem on Computer B. Computer B will fail until it sees full backup set 35 and subsequent log backup set 36 (remember that there

must always be one subsequent complete log backup for a set to be restored). When sets 35 and 36 arrive on computer B, it will repair itself using 35. After the repair is complete, computer A and B will be in sync.

See Also

[Troubleshooting Log Shipping](#)

Advanced Information About Backup and Restore

The topics listed here describe the backup and restore processes in more detail and are intended to be used by advanced users with a thorough understanding of BizTalk Server.

- For information about marked transaction, full backups, and logs, see [Marked Transactions, Full Backups, and Log Backups](http://go.microsoft.com/fwlink/?LinkId=151565) (http://go.microsoft.com/fwlink/?LinkId=151565).
- For information about BizTalk Server log shipping, see [BizTalk Server Log Shipping](http://go.microsoft.com/fwlink/?LinkId=151566) (http://go.microsoft.com/fwlink/?LinkId=151566).
- For information about scheduling backup BizTalk Server job, see [How to Schedule the Backup BizTalk Server Job](http://go.microsoft.com/fwlink/?LinkId=151568) (http://go.microsoft.com/fwlink/?LinkId=151568).
- For information about backing up custom databases, see [How to Back Up Custom Databases](http://go.microsoft.com/fwlink/?LinkId=151569) (http://go.microsoft.com/fwlink/?LinkId=151569).
- For information about creating a linked server, see [How to Create a Linked Server](http://go.microsoft.com/fwlink/?LinkId=151570) (http://go.microsoft.com/fwlink/?LinkId=151570).
- For information about viewing the history of restored backups, see [Viewing the History of Restored Backups](http://go.microsoft.com/fwlink/?LinkId=151572) (http://go.microsoft.com/fwlink/?LinkId=151572).

In This Section

- [BizTalk Server Log Shipping Using a Windows Cluster Name and IP Address](#)
- [Restoring Production from a Warm Backup](#)

BizTalk Server Log Shipping Using a Windows Cluster Name and IP Address

It is possible to simplify BizTalk Server log shipping by using two instances of a SQL Server cluster as the source and destination servers in a BizTalk Server log shipping scenario. Then, in the event of a disaster recovery event, database recovery is simplified by merely switching the name and IP address resources associated with the clustered SQL Server instances as described below. When using this approach there is no need to run the UpdateDatabase.vbs

script as described in the topic [How to Restore Databases in the Backup BizTalk Server Job](#) because the database name is unchanged.



Note

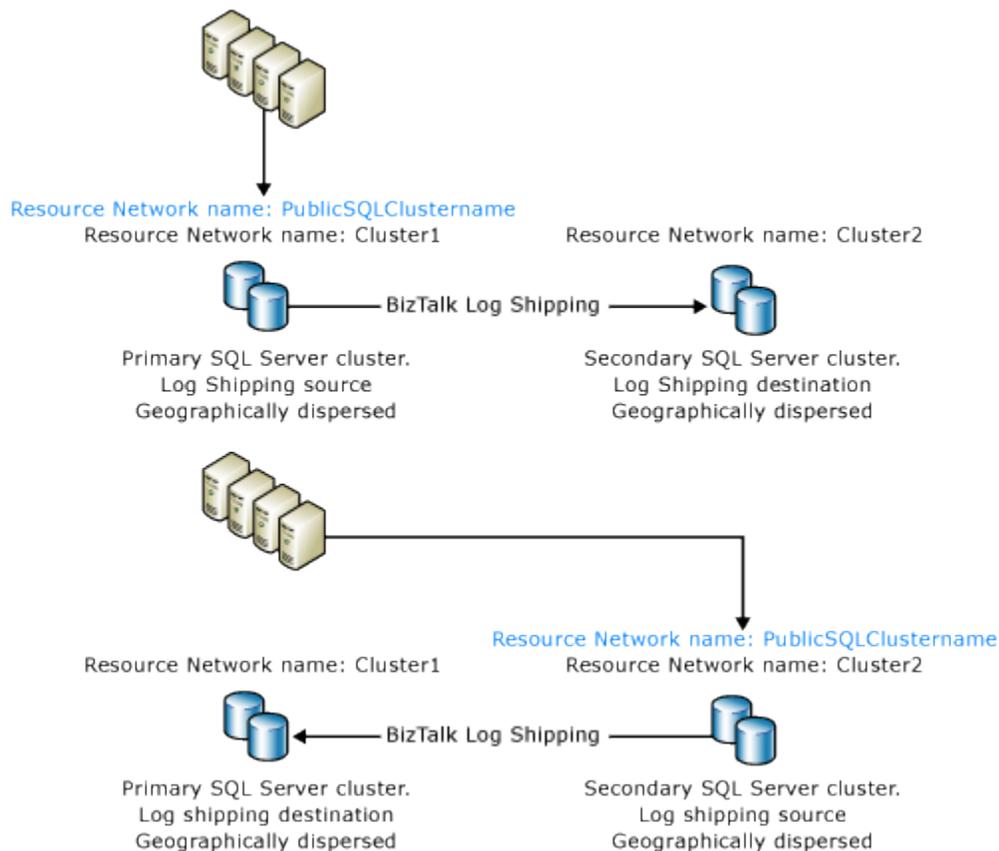
To increase fault tolerance for the clustered SQL Server instances, the clustered SQL Server instances should be geographically separated.

▶ **To implement BizTalk Server log shipping using a Windows Server Cluster name and IP address resource**

1. Stop the production BizTalk servers.
2. Perform a BizTalk Server log shipping restore to the secondary SQL Server cluster.
3. Follow the steps described in the topic [Configuring BizTalk Server Log Shipping](#) to reconfigure BizTalk Server log shipping so that the secondary SQL Server cluster instance is now the source group and the primary SQL Server cluster instance is now the destination group.
4. Stop the IP and network name resource PublicSQLClustername on the primary SQL Server cluster instance.
5. Configure and start the PublicSQLClustername IP and network name cluster resources on the secondary SQL Server cluster instance.
6. Start the production BizTalk servers.
7. Verify the log-shipping restore.
8. Start BizTalk Server related services on the production site.

After performing these steps, the BizTalk group is pointing to the secondary SQL Server cluster instance as illustrated in the following figure.

The following figure illustrates how to configure BizTalk Server log shipping by using two clustered instances of SQL Server and moving the clustered name and IP address resource.



BizTalk Server Log Shipping implementation using Windows Server cluster name and IP address resources

See Also

[High Availability for Databases](#)

[Configuring BizTalk Server Log Shipping](#)

[How to Restore Databases in the Backup BizTalk Server Job](#)

Restoring Production from a Warm Backup

If the source system becomes unreliable, it is possible to restore the databases from the destination to the source. Perform the following procedure to restore databases from the destination to the source.

► To restore the databases from the destination to the source follow these steps

1. Disable all backup jobs on the source (production) SQL Server instance(s).

2. Wait for all restore jobs to complete on the destination disaster recovery (DR) SQL Server instance(s).
3. Disable all restore jobs on the destination (DR) SQL Server instance(s).
4. Restore all databases with STOPATMARK on the destination (DR) SQL Server instance(s).
5. Stop all BizTalk Server services on all BizTalk servers.
6. Drop all BizTalk Server-related databases on the source (production) SQL Server instance(s).
7. Back up (full) all databases on the destination (DR) SQL Server instance(s).
8. Restore (full) all BizTalk Server databases backed up in step 7 to the source (production) SQL Server instance(s).
9. Restart all BizTalk servers including the master secret server.
10. Drop all BizTalk Server-related databases on the destination (DR) SQL Server instance(s).
11. Enable backup jobs on the source SQL Server instance(s).
12. Enable restore jobs on the destination (DR) SQL Server instance(s).

See Also

[Advanced Information About Backup and Restore](#)