

# Microsoft® Business Solutions CRM Integration for Great Plains: Performance Results for Record Transfers

**White Paper**

**Published: October 2003**

**MICROSOFT CORPORATION**

#

i

The information contained in this document represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication.

This White Paper is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT.

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.

© 2003 Microsoft Corporation. All rights reserved.

Microsoft, Active Directory, BizTalk, and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

The names of actual companies and products mentioned herein may be the trademarks of their respective owners.

**Table of Contents**

- Microsoft® Business Solutions CRM Integration for Great Plains: Performance Results for Record Transfers..... i**
- Overview..... 1**
  - Test System Configuration Information.....1
  - Settings and Administration Tool Configuration .....2
  - BizTalk Server Configuration .....2
  - Data Set Information.....2
  - Assumptions.....2
  - Records Per Data Set .....3
- Performance Results..... 3**
- Existing Performance Optimizations ..... 4**
  - Existing Optimizations for BizTalk Server.....4
  - Existing Optimization for Communication.....4
- Performance Recommendations..... 5**
  - General Performance Recommendations .....5
  - Optimizing BizTalk Server .....5
    - Optimizing Specifications.....5
    - Optimizing Maps .....6
    - Optimizing the Messaging Cache .....6
    - Optimizing Server Properties.....7
    - Optimizing Registry Settings .....7
    - Architecture Design Considerations .....8
  - Optimizing SQL Server .....9
    - Optimizing Server Performance.....9
- Evaluating the Performance of Your Own Configuration..... 11**
  - Improving the Architecture.....14
  - Maintaining Performance.....14



# Overview

Understanding the various components of Microsoft® Business Solutions CRM Integration for Great Plains enables you to create a high-performance system. There are several methods, such as identifying potential bottlenecks, addressing latency, adding hardware, and managing databases, that you can use to increase the performance of Microsoft CRM Integration for Great Plains. However, the extent to which you can achieve an optimal solution depends on the complexity of your organization's system architecture and your budget requirements.

To meet your transaction requirements and enhance performance, you must optimize your Microsoft CRM Integration for Great Plains system. When we discuss this system, we are referring to the operating system (either Microsoft Windows® 2000 Server or Advanced Server), and the software that makes up the system including Microsoft CRM Integration for Great Plains, Microsoft SQL Server™ 2000 Desktop Engine (MSDE 2000), and Microsoft BizTalk® Server 2002. As you will see, optimizing your operating system and database interactions will enhance performance the most. Therefore, this paper focuses primarily on how to optimize these elements.

This document provides you with some test data results as a guide for where to start given your environment. To gather these results, the Microsoft CRM Integration team configured a system, tested three different sizes of data sets against the configuration, and then determined what modifications to make to the configuration in order to achieve greater performance results.

You can use this information as a baseline for your own data set. However, for the most accurate results, you should configure your own system, test your data set against it, and then evaluate the results to determine if your initial configuration can meet your transaction requirements. It is likely that you will need to reconfigure your architecture to achieve optimal performance. After each reconfiguration, test it again and evaluate the results. Once you have achieved the results you want, you should initiate a maintenance plan, because server performance changes over time, and the quantity and type of transactions that your organization handles might change.

## Test System Configuration Information

This section describes the hardware and software configurations of the Microsoft CRM Integration for Great Plains systems used for performance testing. For all configurations, Microsoft Windows 2000 Server was used as the operating system. All systems met the software requirements specified in the *Microsoft Business Solutions CRM Integration for Great Plains Guide*.

Computer 1	Computer 2	Computer 3
<p>Hardware</p> <ul style="list-style-type: none"><li>• Intel Pentium 4 with 1.7-gigahertz (GHz) CPU</li><li>• 1 gigabyte (GB) of physical RAM</li><li>• 76-GB hard disk drive (30/46)</li></ul>	<p>Hardware</p> <ul style="list-style-type: none"><li>• Intel Pentium 4 with 1.7-GHz CPU</li><li>• 1 GB of physical RAM</li><li>• 76-GB hard disk drive (30/46)</li></ul>	<p>Hardware</p> <ul style="list-style-type: none"><li>• Intel Pentium 4 with 1-GHz CPU</li><li>• 500 MB of physical RAM</li></ul>
<p>Software</p> <ul style="list-style-type: none"><li>• Microsoft SQL Server</li></ul>	<p>Software</p> <ul style="list-style-type: none"><li>• Microsoft CRM</li><li>• Microsoft CRM Integration for Great Plains</li><li>• Microsoft BizTalk Server 2002</li><li>• SQL Server 2000 Desktop</li></ul>	<p>Software</p> <ul style="list-style-type: none"><li>• Active Directory®</li></ul>

	Engine • Document Exchange	
--	-------------------------------	--

## Settings and Administration Tool Configuration

The only setting that was modified to affect the performance testing was the **Time To Live** option, which was changed to **999999999** to help ensure that messages did not time out when large amounts of data were being processed. After the record transfer, this number was changed back to its default setting of **43200**. Other than that, all required fields were set, as well as default **Shipping Methods**, **Payment Terms**, and an **Account Category Code**. In addition, **Accounts** was selected from the **Create customers as** drop-down list.

## BizTalk Server Configuration

For these tests, most BizTalk Server configurations were not modified. The original configurations for receive functions, organizations, document definitions, maps, envelopes, messaging ports, and channels that were created during setup existed during these performance tests. However, BizTalk Document Tracking was disabled prior to performing any testing.

## Data Set Information

The performance tests were based on three different sizes of data sets. On average, messages were small (ranging in size from 1 KB to 3 KB). Performance testing was not completed on orders being transmitted from Microsoft Great Plains to Microsoft CRM or vice versa. The data used was modeled after real-world customer relationship management scenarios. The same configuration was used for each data set. The following provides details regarding data sizes:

- Microsoft Great Plains customer records ranged from 1 kilobyte (KB) to 3 KB.
- Microsoft Great Plains address records ranged from 1 KB to 3 KB.
- Microsoft Great Plains item records ranged from 1 KB to 3 KB.
- Microsoft Great Plains pricing-related data was typically 1 KB.
- Microsoft Great Plains currency records ranged from 1 KB to 3 KB.
- Microsoft CRM account, contact, and address records were less than 2 KB.

## Assumptions

The following assumptions were applied to the data-set sizes based on historical customer data.

	Customers	Addresses	Items	Currencies	Price levels	Discount types
<b>Data Set 1</b>	Fewer than 1,000	Fewer than 2,000	Fewer than 200	1	1 - 5	1
<b>Data Set 2</b>	1,000 – 10,000	2,000 – 50,000	200 – 700	2 - 5	6 - 20	2 - 5
<b>Data Set 3</b>	More than 10,000	More than 50,000	More than 700	More than 5	More than 20	More than 5

With regard to the data in the previous table, the team worked with one customer that had 100,000 customer records and another with 200,000. Each customer record typically had one or more addresses associated with it. There are some customers that use Microsoft CRM with Microsoft Great Plains and have over 8,000 products

listed; however, the products are raw goods and typically customers do not want these records transferred to Microsoft CRM and used as products on orders. This means that of the 8,000 products, only 700 or so are actually products that can be ordered. In addition, we do realize that there are some larger customers with thousands upon thousands of items.

## Records Per Data Set

The following table lists the number of records per data set.

	Data set 1	Data set 2	Data set 3
<b>Customer</b>	6,490	2,426	4,154
<b>Address</b>	6,733	6,352	4,154
<b>Item</b>	5,966	5,679	14,017
<b>Price Level</b>	1	28	55
<b>Price List</b>	5,951	1,011,429 (112,381 used)	784,952
<b>Currency</b>	US	9	

## Performance Results

The following test results are based on the configurations described previously in "Test System Configuration Information." Performance was measured using SQL Server scripts to determine the difference between when the first and last records were created in Microsoft CRM. The numbers listed in the tables are the result of a one-time record transfer from Microsoft Great Plains to Microsoft CRM.

The following tables list the performance test results.

### Data Set 1

Records	Number of rows	Total time (hr:min:sec)	Records per minute
Customer	6,490	00:47:16	137.29
Address	6,733	00:28:32	236.00
Item	5,966	00:20:11	295.64
Pricing (part 1)	5,951	00:30:14	196.86
Pricing (part 2)	5,951	00:21:36	275.51

### Data Set 2

Records	Number of rows	Total time (hr:min:sec)	Records per minute
Customer	2,426	00:16:43	145.09
Address	6,352	00:21:32	295.03
Item	5,679	00:33:11	171.52
Pricing (part 1)	112,381	11:41:12	160.27
Pricing (part 2)	112,381	6:51:19	273.22

### Data Set 3

Records	Number of rows	Total time (hr:min:sec)	Records per minute
Customer	4,154	00:45:00	92.31
Address	4,154	01:42:00	40.73
Item	14,017	01:02:00	226.08
Pricing (part 1)	784,952	6 days, 4 hours	88.36
Pricing (part 2)	784,952	8 days, 6 hours	66.07

#### Notes

- You can process documents using non-transactional message queues, which would result in higher performance, but we do not recommend it. If you use non-transactional message queues, you might lose data. For example, if the server or transport fails, you cannot recover the lost data.
- You might experience fewer records per minute if your network has a heavy traffic load.
- These numbers reflect only the transfer of records from Microsoft Great Plains to Microsoft CRM.
- The performance tests were conducted on an out-of-the-box setup, with no customizations to Microsoft Great Plains, Microsoft CRM, or Microsoft CRM Integration for Great Plains. Customization, such as adding more fields to be integrated, may have an effect on performance.

## Existing Performance Optimizations

Several existing conditions already help optimize the Microsoft CRM Integration for Great Plains system. The list below identifies the “out-of-the-box” BizTalk Server configuration, which was not modified to help maintain optimal performance of Microsoft CRM Integration for Great Plains. To further optimize your system, you can apply the suggestions in the "Performance Recommendations" section that follows.

### Existing Optimizations for BizTalk Server

The following have already been configured to help optimize performance of BizTalk Server, which in turn optimizes performance of Microsoft CRM Integration for Great Plains:

- **Configured a receive function (such as Message Queuing or File).** Using receive functions provides better performance than using the **Submit** method to call the **Interchange** object because the receive functions run within BizTalk Services and can cache the internal state of objects. Whereas, the **Submit** method runs out-of-process and has to rebuild its internal state for each call.
- **Disabled document tracking.** Tracking is disabled to minimize the number of reads and writes to the database for a single transaction. Therefore, this “out-of-the-box” configuration eliminates certain functionality in exchange for better performance. If you enabled the **Tracking** field, BizTalk Server connects to the database and logs data, which will have a slight affect on performance. However, if tracking is enabled and if the tracking interchanges size limit is exceeded, the performance of BizTalk Server is greatly affected.
- **Used HTTP, Message Queuing, and File transport services.** These services are used to write data to disk drives on servers other than the local BizTalk server.

### Existing Optimization for Communication

Microsoft CRM Integration for Great Plains optimizes BizTalk Server to take advantage of asynchronous communication, which is highly scalable and provides a high level of throughput. In addition, by optimizing for

asynchronous communication, the following submissions, which would fail if using synchronous communication, succeed:

- If the transaction criteria matches more than one channel.
- If the interchange contains multiple documents.

# Performance Recommendations

This section provides general recommendations for optimizing system settings and includes topics for optimizing Windows 2000, BizTalk Server, and SQL Server settings.

## General Performance Recommendations

### Optimizing Microsoft Windows 2000 Server and Microsoft Windows 2000 Advanced Server settings

Apply best practices, such as not running unnecessary services or protocols, to improve Windows 2000 performance. Many techniques used to optimize Windows 2000 Server and Windows 2000 Advanced Server also can be used to optimize BizTalk Server and Microsoft CRM Integration for Great Plains. For more information about optimizing Microsoft Windows 2000 settings, see "Best Practices" in the "Installing Windows 2000 Server" book in Windows 2000 Help or for Microsoft Windows 2000 Advanced Server settings, see "Best Practices" in the "Installing Windows 2000 Advanced Server" book in Windows 2000 Advanced Server Help.

### Optimizing network throughput

Maintain fast, reliable network connectivity between transport services, BizTalk Messaging Services, and the databases (100 megabits per second or higher Ethernet). To optimize network throughput, use multiple adapters in each server, with a unique switch port for each, with inbound and outbound transactions separated between the network interface cards (NICs). When used in conjunction with Microsoft Windows 2000 Network Load Balancing or Microsoft Application Center 2000 component load balancing (IInterchange or IPipelineComponent), performance is significantly increased.

## Optimizing BizTalk Server

The following sections provide guidelines for optimizing BizTalk Server.

### Optimizing Specifications

When using existing specifications or when designing your own specifications, you can configure validation rules within a specification. For example, you can specify that a field contain a particular data type (such as a string) or you can specify more complex rules (such as requiring a field to be validated against a list of 80 values). The latter requires that the data be checked against 80 values and, if it does not conform to even one of them, the specification fails validation. Specification validation in BizTalk Server is enabled by default; it affects performance but ensures the validity of the data being sent or received.

To improve performance, you can disable validation, which might not adversely affect the validity of data because BizTalk Server has several other mechanisms for validating data. For example, if you have a map that requires certain data types and values and specification validation is disabled, the data successfully passes the parsing phase but fails during the serializing phase when the map is applied. You might also choose to disable validation because you have complete control of how the data is sent (such as application-to-application transactions).

You can use the **NoValidation** registry key to disable validation. If specification validation is not required (documents are not validated against a specification for settings such as minimum/maximum values, data types, and/or required values), set this to a nonzero value and the data will not be validated. This might help with performance but can result in non-valid data being sent. Changing the specification validation to a nonzero

value might be the right option for servers that are receiving documents for which the organization has control over the structure.

### Caution

- Incorrectly editing the registry may severely damage your system. Using the registry editors to modify the registry bypasses the standard safeguards that are provided by administrative tools, allowing settings that could degrade performance, damage your system, or even require you to reinstall the Windows operating system. If you must edit the registry directly, first back it up. See Registry Editor Help or "Backing Up the Windows 2000 Registry" in the Microsoft Windows 2000 Server Resource Kit for more information.

## Optimizing Maps

Translating data, as part of a transaction, might reduce the performance of BizTalk Server. Mapping specifications can be CPU-intensive, which can reduce the ability of BizTalk Server to process the overall transaction. The level to which BizTalk Server is affected depends largely on the complexity of the map being used for translation. For example, mapping a field called **productID** in the source specification to a field called **itemID** in the destination specification is not as intensive as performing a complex mathematical operation on the data being translated.

### Important

- The recommendations in this section should be applied against any new fields that you map. You should be very cautious about applying these performance recommendations against existing mappings, particularly mappings that involve functoids, as it may result in data loss or corruption.

### Recommended solutions for designing maps

The following list provides optimization guidelines for designing maps:

- Use functoids only when needed. Functoids use script, which causes BizTalk Server to load a scripting engine. This might degrade performance as opposed to native XML transformations.
- Use the Database Lookup functoid only when needed. BizTalk Server must establish a database connection, query for data, populate a recordset, and close the connection each time this functoid is used. This can degrade performance of BizTalk Server.
- Avoid invoking COM objects within functoid scripts and using custom functoids. Both of these techniques cause BizTalk Server to instantiate an instance of a COM object. This affects the performance of the overall operation. In addition, the performance of instantiating COM objects is affected by the answers to the following questions:
  - Is it in-process or out-of-process?
  - Is the object local or remote?
  - How well is the object written?
  - Is it transactional?
- Analyze each map to determine how to achieve optimal performance. For example, two different techniques can be used to concatenate two source fields together and have the resultant value placed in two different destination fields. The first concatenation technique would be to use one concatenation functoid and map its output to two places. The second concatenation technique would be to use two functoids, each with a single output link to a destination source field. The first technique is slightly more efficient because the script, which runs as part of the functoid, is called only once. Through testing, you can determine which techniques result in the best performance.

## Optimizing the Messaging Cache

In BizTalk Server Administration, you can specify the **Messaging Management object cache refresh interval (seconds)** property in the **BizTalk Server Group Properties** dialog box. You can set this box to a maximum of 300 seconds. BizTalk Server 2002 caches configurations (such as channels, messaging ports, envelopes, and document definitions) in memory to avoid calling the database each time. If these objects are not regularly changed, set this value to 300 to reduce the number of times data is written to and read from the database. Because BizTalk Server refreshes management objects every five minutes, the BizTalk Messaging Service must be restarted for the change to take effect immediately.

## Optimizing Server Properties

In BizTalk Server Administration, you can set the following properties in the **Properties** dialog box for each server:

- **Maximum number of receive function threads allowed.** You can specify how many receive function worker threads per processor you want for a receive function. Setting this too low can cause a slowdown in BizTalk Server because it uses I/O completion ports. Setting this too high should not have serious effects, but it might cause performance degradation. You can adjust this number to find the optimal value for your setup. The recommended value for the **Maximum number of receive function threads allowed** property is 4.
- **Maximum number of worker threads allowed.** You can specify the number of worker threads per processor for the processing side. (Note that the Partner Edition of BizTalk Server supports only one processor.) By appropriately adjusting the number of worker threads, you can improve performance. In BizTalk Server Administration, right-click a server in the console tree and click **Properties**. Change the default value in the **Maximum number of worker threads per processor allowed** box. The default value is 4. The recommended value is from 10 to 16, depending on the deployment.

## Optimizing Registry Settings

You can optimize the registry settings to improve BizTalk Server performance. All keys should be added as DWORD values to `\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\BTSSVC`. To improve performance, you can implement the following registry setting adjustments:

- **NoValidation.** Use this registry key to disable specification validation. If specification validation is not required (documents are not validated against a specification for minimum/maximum values, data types, or required values, for example), set this to a nonzero value and the data will not be validated. While this might improve performance, it can result in nonvalid data being sent. This setting would be appropriate for servers that are receiving documents for which the organization controls the structure.
- **ParserRefreshInterval.** By default this is set to 60,000 (60 seconds). This value indicates how often BizTalk Server should check the database to see if a new parser has been added. (This is the only group-level property that is refreshed while the server is running.) If no new parsers will be added, set this value to 0 and BizTalk Server will not check the database. This value is also used to verify if new parsers have been added to the Tracking database group settings or if the settings have been altered.
- **CacheSize.** Use this registry key to indicate how large the BizTalk Server management object cache is allowed to grow. The default value for this is 20. Therefore, as soon as the BizTalk Server object cache exceeds 20 channels, messaging ports, envelopes, or document definitions, BizTalk Server must delete some items from memory. If a server has a high volume of memory, this value can be set higher (above 20) and BizTalk Server will keep more in memory. This does not affect the refresh interval. BizTalk Server deletes the cached objects and reloads them when they have expired.
- **BatchSize.** Use this value only with the Message Queue receive function. By default **BatchSize** is set to 20. To improve performance, BizTalk Server reads up to 20 items at a time from the queue and sends all 20 within one transaction. Reducing the number of times data is written to a database greatly improves performance. If a deadlock occurs, and BizTalk Server ends the transaction, it must resubmit the items.

BizTalk Server does not lose the documents. Values that exceed 20 have not been tested. Do not set this value to 0.

### Caution

- Incorrectly editing the registry may severely damage your system. Using the registry editors to modify the registry bypasses the standard safeguards that are provided by administrative tools, allowing settings that could degrade performance, damage your system, or even require you to reinstall the Windows operating system. If you must edit the registry directly, first back it up. See Registry Editor Help or "Backing Up the Windows 2000 Registry" in the Microsoft Windows 2000 Server Resource Kit for more information.

### To improve processing performance

1. On the **Start** menu, click **Run**, type **regedit**, and then click **OK**.
2. In the **Registry Editor** dialog box, click the expand indicator (+) for the **HKEY\_LOCAL\_MACHINE** node; expand **SYSTEM**, expand **CurrentControlSet**, and expand **Services**, and then click **BTSSvc**.
3. Right-click in the details pane, point to **New**, and then click **DWORD Value**.
4. Type **NoValidation** and press ENTER twice.
5. In the **Edit DWORD Value** dialog box, in the **Value data** box, type any nonzero value, such as 1, and click **OK**.

The default value is 0 (validation).

6. Right-click in the details pane, point to **New**, and then click **DWORD Value**.
7. Type **ParserRefreshInterval** and press ENTER.
8. Right-click in the details pane, point to **New**, and then click **DWORD Value**.
9. Type **CacheSize** and press ENTER twice.
10. In the **Edit DWORD Value** dialog box, in the **Value data** box, type the number of channels, messaging ports, envelopes, and document definitions that you expect to or do have in memory, and click **OK**.
11. Right-click in the details pane, point to **New**, and then click **DWORD Value**.
12. Type **BatchSize** and press ENTER twice.
13. In the **Edit DWORD Value** dialog box, in the **Value data** box, type a value for the number of items you want to process as a batch, and click **OK**.

## Architecture Design Considerations

To successfully plan an initial architecture, you must consider the volume of transactions your system needs to handle. If your configuration contains fewer than three computers, it is assumed that you have an extremely low volume of transactions. A three-server configuration with a single SQL Server license is based on a low to moderate volume of transactions. A three-server configuration with two SQL Server licenses is based on a moderate to high volume of transactions. If you know your system will need to handle a high volume of transactions quickly, then you will want to invest in a six-server configuration.

Depending on the server configuration, the following outlines how to distribute the required software on each computer:

- Create a three-server configuration with a single SQL Server license as follows:
  - Install Microsoft CRM, Microsoft CRM Integration for Great Plains, and BizTalk Server on one computer.
  - Install SQL Server and have it hosting the Microsoft CRM, Microsoft CRM Integration for Great Plains, BizTalk Server, and Microsoft Great Plains databases.

- Install Active Directory on one computer.
- Create a three-server configuration with two SQL Server licenses as follows:
  - Install Microsoft CRM on one computer with SQL Server hosting the Microsoft CRM databases.
  - Install Microsoft CRM Integration for Great Plains and BizTalk Server on one computer with SQL Server hosting Microsoft CRM Integration for Great Plains, BizTalk Server, and Microsoft Great Plains databases.
  - Install Active Directory on one computer.
- Creating a six-server configuration as follows:
  - Install Microsoft CRM and SQL Server on separate computers (two separate computers used).
  - Install Microsoft Great Plains on a separate computer that is also running SQL Server (one computer used).
  - Install Microsoft CRM Integration for Great Plains and SQL Server on separate computers (two separate computers used).
  - Install Active Directory on one computer.

## Optimizing SQL Server

The following sections highlight ways in which you can optimize your Microsoft SQL Server database and servers to maximize throughput of your data between Microsoft Great Plains and Microsoft CRM.

### Optimizing Server Performance

Microsoft SQL Server 2000 automatically tunes many of the server configuration options, therefore requiring little, if any, tuning by a system administrator. Although these configuration options can be modified by the system administrator, it is generally recommended that these options be left at their default values, allowing SQL Server to automatically tune itself based on run-time conditions.

However, if necessary, the following components can be configured to optimize server performance:

- SQL Server Memory
- I/O subsystem

### Optimizing Server Performance Using Memory Configuration Options

The memory manager component of SQL Server 2000 eliminates the need for manual management of the memory available to SQL Server. When SQL Server starts, it dynamically determines how much memory to allocate based on how much memory the operating system and other applications are currently using. As the load on the computer and SQL Server changes, so does the memory allocated.

The following server configuration options can be used to configure memory usage and affect server performance:

- **min server memory**
- **max server memory**
- **max worker threads**
- **index create memory**
- **min memory per query**

The **min server memory** server configuration option can be used to ensure that SQL Server does not release memory below the **min server memory** value after that value has been reached. This configuration option can

be set to a specific value based on the size and activity of your SQL Server. If you choose to set this option, you must leave enough memory for the operating system and other programs. If the operating system does not have enough memory it will request memory from SQL Server, affecting SQL Server performance.

The **max server memory** server configuration option can be used to specify the maximum amount of memory that SQL Server can allocate when it starts and while it runs. This configuration option can be set to a specific value if you know there are multiple applications running at the same time as SQL Server and you want to guarantee that these applications have sufficient memory to run. If these other applications, such as Web or e-mail servers, request memory only as needed, then do not set the **max server memory** server configuration option, because SQL Server will release memory to them as needed. However, applications often use whatever memory is available when they start and do not request more if needed. If an application that behaves in this manner runs on the same computer at the same time as SQL Server, set the **max server memory** server configuration option to a value that guarantees that the memory required by the application is not allocated by SQL Server.

Do not set **min server memory** and **max server memory** server configuration options to the same value, thereby fixing the amount of memory allocated to SQL Server. Dynamic memory allocation will give you the best overall performance over time.

The **max worker threads** server configuration option can be used to specify the number of threads used to support the users connected to SQL Server. The default setting of 255 can be slightly too high for some configurations, depending on the number of concurrent users. Because each worker thread is allocated, even if it is not being used (because there are fewer concurrent connections than allocated worker threads), memory resources that can be better utilized by other operations, such as the buffer cache, can be unused. Generally, this configuration value should be set to the number of concurrent connections, but cannot exceed 32727. Concurrent connections are not the same as user login connections. The pool of worker threads for an instance of SQL Server only needs to be large enough to service the number of user connections that are actively executing batches at the same time in that instance. Increasing the number of worker threads beyond the default value may result in negative server performance.

The **index create memory** server configuration option controls the amount of memory used by sort operations during index creation. Creating an index on a production system is usually an infrequently performed task, often scheduled as a job to execute during off-peak time. Therefore, when creating indexes infrequently and during off-peak time, increasing this number can improve the performance of index creation. Keep the **min memory per query** configuration option at a lower number, however, so the index creation job will still start even if all the requested memory is not available.

The **min memory per query** server configuration option can be used to specify the minimum amount of memory that will be allocated for the execution of a query. When there are many queries executing concurrently in a system, increasing the value of the **min memory per query** can help improve the performance of memory-intensive queries, such as substantial sort and hash operations. However, do not set the **min memory per query** server configuration option too high, especially on very busy systems, because the query will have to wait until it can secure the minimum memory requested or until the value specified in the **query wait** server configuration option is exceeded. If more memory is available than the specified minimum value required to execute the query, the query is allowed to make use of the additional memory, provided that the memory can be used effectively by the query.

## Optimizing Server Performance Using I/O Configuration Options

The recovery interval server configuration option can be used to configure I/O usage and affect server performance. The recovery interval server configuration option controls when SQL Server issues a checkpoint in each database. By default, SQL Server determines the best time to perform checkpoint operations. However, to determine if this is the appropriate setting, monitor disk write activity on the database files using Microsoft Windows System Monitor. Spikes of activity, which cause disk utilization to reach 100 percent, can affect performance. Changing this parameter to cause the checkpoint process to occur less often can improve overall performance in this situation. However, continue to monitor performance to determine if the new value has had

a positive effect on performance. For more information, see "recovery interval Option" in SQL Server Books Online.

# Evaluating the Performance of Your Own Configuration

You can use Microsoft Windows 2000 System Monitor to test the performance of BizTalk Messaging Services.

To use System Monitor, on the **Start** menu, point to **Settings**, click **Control Panel**, double-click **Administrative Tools**, and then double-click **Performance**. System Monitor graphically displays counter readings as they change over time. There are, however, different counters that should be monitored depending on the system component being monitored. Numerous white papers are available describing how to monitor performance for Windows 2000, Internet Information Services (IIS), SQL Server, and Message Queuing; however, the following table lists only some of the primary objects and counters to monitor. In addition, the table contains specific information regarding the objects and counters to monitor to determine BizTalk Messaging Services performance.

<b>Object</b>	<b>Counter</b>	<b>Observation</b>	<b>Component affected</b>
<b>Network Segment</b>	Bytes received per second/Bytes sent per second	If this number is close to the capacity of the connection, and processor and memory use are moderate, the connection might affect performance.	All
<b>Memory</b>	Available bytes	Available bytes should not stay below 10 megabytes (MB) consistently. If so, a memory spike would cause paging to disk to start.	All
	Page Faults/sec, Memory: Pages Input/sec, and Memory: Page Reads/sec	If these numbers are low, the server should be responding to requests quickly. If they are high, an increase in the amount of RAM on your server might be needed.	All
<b>Physical Disk</b>	Disk read/writes/sec	Combined, these two counters should be significantly under the maximum capacity for the disk device. To enable this counter, on the <b>Start</b> menu, point to <b>Programs</b> , point to <b>Accessories</b> , and then click <b>Command Prompt</b> . At the command prompt, type	SQL Server, Message Queuing, and File transport services

		<b>diskperf -y</b> and press ENTER. Then restart the computer.	
	% Disk time	This counter should be well below 100 percent. If it is above this value (and it can go into the 1000 percent range), add more physical disks or move one of the databases to another server.	SQL Server, Message Queuing, and File transport services
	Current Disk Queue Length	This counter is the number of requests outstanding on the disk at the time the performance data is collected. This counter should average less than 2 for good performance.	BizTalk Server SQL Server
<b>SQL Server</b>	I/O transactions/sec	Indicates how much activity the computer running SQL Server actually performs.	SQL Server
<b>BizTalk Server</b>	Documents Processed/sec	Indicates how quickly BizTalk Server 2002 is polling documents from its Work queue and sending them.	BizTalk Messaging Services
<b>BizTalk Server</b>	Documents Received/sec	Indicates how quickly BizTalk Server is sending documents to the Work queue. This number reflects only the number of documents BizTalk Server has received (this includes documents that fail parsing), not the number of documents BizTalk Server checkpoints to its Work queue. The number of documents that are checkpointed to the Work queue is essentially equal to the Documents Processed/sec counter.	BizTalk Messaging Services
<b>BizTalk Server</b>	Synchronous Submissions/sec, Asynchronous	Indicates how quickly the <b>Submit</b> method and/or the <b>SubmitSync</b> method calls occur. Because each	BizTalk Messaging Services

	Submissions/sec	interchange can contain any number of documents, this counter is not useful for determining documents processed. If pass-through (processing interchanges without parsing them) is being used exclusively, this is the counter you need to monitor to determine inbound performance.	
<b>Message Queue</b>	Messages in queue	This number should not get extremely large (over 50,000) because it will cause excessive memory use on the Message Queue server and degrade the performance of the entire system.	Message Queuing transport service
<b>System</b>	Processor Queue Length	This counter displays the number of threads waiting to be executed in the queue that is shared by all processors on the system. If this counter has a sustained value of two or more threads, the processor is degrading the performance of the entire system.	All
	Context switches/sec	If this is a high number on BizTalk Server, it could be because send and receive functions are running on the same server. If this is the case, consider separating the send and receive functions to separate servers.	All
<b>Processor</b>	%Processor Time	If this counter's value is high, while the network adapter card and disk I/O remain well below capacity, the processor is affecting performance. On a multiprocessor computer, examine this counter to identify any imbalance. Additionally,	All

while peak utilization can be 100 percent, sustained utilization should be below this value. All server elements can be scaled horizontally.

Data obtained from performance monitoring is also useful for identifying symptoms that can contribute to problems. For example, a high amount of disk activity might indicate that SQL Server is writing a large amount of data to disk, but it might also indicate that the system is often paging to disk. Excessive disk paging typically indicates that memory is too low. In this case, the disk activity is the indicator of a problem, but not the problem itself.

## Improving the Architecture

By identifying areas that affect performance, you can refine the system architecture to achieve optimal results. For example, hardware might need to be upgraded, or different components of BizTalk Server might need to be moved to separate servers. Additionally, you might need to adjust BizTalk Server settings. For more information about adjusting settings, see "Optimizing Server Properties" earlier in this white paper.

Ensure that changes are made methodically. After implementing a change, such as adding CPUs, changing a setting, or separating functionality to separate servers, document the change and then test the new configuration to determine if performance has improved. If the new configuration enhances performance, you can continue to modify the configuration. However, if the new configuration is degrading performance, reconfigure the system to its previous state and analyze the data to determine what might be contributing to the problem.

## Maintaining Performance

To maintain performance, create an ongoing maintenance plan to ensure the health and future performance of BizTalk Server. BizTalk Server most likely will process increased numbers of documents over time. Although the initial BizTalk Server architecture might have performed well, if the number of transactions has increased over time, it might not be capable of sustaining the same level of performance unless the system architecture or configuration is modified. Hardware failures, or out-of-memory conditions, also can affect the performance of a system.

### Creating a Performance Maintenance Plan

A successful monitoring plan includes:

- **A detailed plan of which counters to monitor.** For example, there are several Windows 2000 System Monitor counters, thresholds to monitor, and possible interpretations of the problems that might be indicated by the thresholds. Additionally, you can also monitor application-level problems such as whether SQL Server queries should return valid data or return hardware problem conditions.
- **A regular schedule for running the `int_DatabaseCleanup` stored procedure.** There are two integration tables, `IntegrationMessage` and `IntegrationMessageDetail`, that store messages that have been processed by the Microsoft CRM Integration for Great Plains database. If maintenance is not performed on these tables, they will grow indefinitely until your computer no longer has disk space to handle the messages or other processes. To learn more about maintaining these tables using the `int_DatabaseCleanup` stored procedure, see Chapter 10, "Operating and Maintaining the Microsoft CRM Integration for Great Plains System" in the *Microsoft Business Solutions CRM Integration for Great Plains Guide*.
- **Identifying processes for resolving performance issues.** For example, a Windows 2000 event-log message might be written, e-mail notification might be enabled, or a custom program might start a series of events.

- **Defining which performance thresholds generate a notification.** For example, a monitoring tool might be configured to write an event to the Windows 2000 event log if the CPU reaches 90-percent use capacity. However, if the CPU stays at 90-percent use capacity for more than 5 minutes, an e-mail message might be sent. Defining different actions for different problem-severity levels depends on the monitoring tool that you use.
- **Defining the course of action in response to events published in Windows 2000 event logs.** Windows 2000, BizTalk Server 2002, SQL Server, and native Microsoft transport services such as Message Queuing and HTTP write events to the Windows 2000 event log. This event log contains valuable information that can indicate impending problems on one of the servers in a BizTalk Server system. If addressed early, you can avoid a system failure that would degrade the performance of the overall system. A plan should include the frequency at which the logs are monitored and archived, and include the party responsible for the tasks. A third-party tool can be used to accumulate the logs from multiple servers and write them to a database for consolidated analysis. These tools can also be used to purge the logs from the servers.
- **Determining the required tools for performance monitoring.** The built-in Windows 2000 System Monitor tool examines performance counters and can take action when a predefined condition occurs. For more information about configuring System Monitor alerts, see "Setting up a monitoring configuration" in Windows 2000 Help.