

Performance

Microsoft Dynamics CRM 4.0

Optimizing and Maintaining Performance

White Paper

Date: February 2008



Credits

This document was initiated and developed by the Microsoft Dynamics CRM Engineering for Enterprise (E²) Team, which would like to acknowledge the following for their efforts in helping to ensure the completeness and accuracy of this document, which is intended to support the broader CRM community.

Key Contributors

Brian Bakke (*Microsoft*)
Grant Geiszler (*Microsoft*)
Sean McNellis (*Microsoft*)
MS CRM Performance Team

Technical Reviewers

Peter Simons (*Microsoft*)
Roger Gilchrist (*Microsoft*)
MS CRM Product Development

MS CRM E² Contributors

Amir Jafri, Program Manager
Jim Toland, Content Manager

Note: This paper leverages and updates content previously published in the *Microsoft Dynamics CRM 3.0 Optimization* whitepaper.

Feedback

Please send comments or suggestions about this document to the MS CRM E² Team content feedback alias (entfeed@microsoft.com).

Microsoft Dynamics is a line of integrated, adaptable business management solutions that enables you and your people to make business decisions with greater confidence. Microsoft Dynamics works like and with familiar Microsoft software, automating and streamlining financial, customer relationship and supply chain processes in a way that helps you drive business success.

U.S. and Canada Toll Free 1-888-477-7989

Worldwide +1-701-281-6500

www.microsoft.com/dynamics

Legal Notice

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.

© 2008 Microsoft Corporation. All rights reserved.

Microsoft, MS-DOS, Windows, Windows Server, Windows Vista, Active Directory, SQL Server, Outlook, BizTalk, FRx, Microsoft Dynamics, Microsoft Dynamics Logo, SharePoint, Visual Basic, Visual C++, Visual SourceSafe, and Visual Studio are either registered trademarks or trademarks of Microsoft Corporation, FRx Software Corporation, or Microsoft Business Solutions ApS in the United States and/or other countries. Microsoft Business Solutions ApS and FRx Software Corporation are subsidiaries of Microsoft Corporation.

All other trademarks are property of their respective owners.

Microsoft

Contents

Preface	5
Introduction to Optimizing and Maintaining Microsoft Dynamics CRM 4.0	6
Server Architecture for Implementing Microsoft Dynamics CRM 4.0	6
The Client Tier.....	7
The Application Tier	7
The Data Tier	7
Recommended Approach to Optimization and Maintenance	8
Optimizing and Maintaining Client Tier Performance	9
Optimizing Network Performance.....	9
Configuring Client-Side Caching.....	9
Configuring Content Expiration	10
Using Compression Techniques.....	10
Accessing Dynamics CRM by using Terminal Services/Citrix	10
Optimizing and Maintaining Microsoft Dynamics CRM for Outlook.....	10
Hardware Requirements	10
Optimizing the Synchronization Process.....	11
Optimizing Address Book Performance	11
Bypassing the Proxy Server for On-premise Deployments within a LAN	12
Improving Startup Time for Microsoft Dynamics CRM for Outlook.....	12
Disabling Extraneous Processes and Applications	13
Reducing Exchange Traffic by Modifying Outlook Rules	13
Optimizing Offline Synchronization.....	14
Microsoft Dynamics CRM for Outlook Diagnostics	15
Optimizing and Maintaining Application Tier Performance	16
Optimizing and Maintaining Microsoft Windows Server	16
Optimizing the Performance of Microsoft Windows Server	16
Monitoring the Performance of Microsoft Windows Server	17
Optimizing and Maintaining the Microsoft .NET Framework and Microsoft .NET Applications	17
Optimizing the Performance of the Microsoft .NET Framework	17
Monitoring the Performance of Microsoft .NET Framework Applications	18
Optimizing and Maintaining Internet Information Services	20
Optimizing the Performance of Internet Information Services.....	20
Monitoring the Performance of Internet Information Services	21
Optimizing the Performance of Microsoft Dynamics CRM 4.0	21
Installing Microsoft Dynamics CRM on Multiple Servers.....	21
Implementing Server Roles.....	22
Optimizing the Performance of Microsoft Dynamics CRM 4.0 Applications.....	22
Optimizing and Maintaining Microsoft Dynamics CRM Reporting Services	25
Optimizing Report Performance	26
Optimization Guidelines.....	26
Optimization Techniques.....	26

Optimizing Workflow	27
Optimizing and Maintaining Data Tier Performance.....	30
Optimizing and Maintaining Microsoft SQL Server and the Microsoft Dynamics CRM Database.....	30
Configuring the Disk Subsystem for Optimal Performance.....	30
Minimizing Locking Contention by Using Isolation Levels.....	31
Disabling Support for Parallel Plan Queries.....	32
Using Efficient Queries	32
Optimizing and Maintaining Query Performance.....	33
Optimizing and Maintaining the Microsoft Dynamics CRM Database	34
Optimizing and Maintaining the CRM E-mail Router.....	35
Optimizing the CRM E-mail Router	35
Maintaining the CRM E-mail Router	36
Appendix A: Additional Resources	37
Information Resources	37
Technical Services	37
Appendix B: CRM E-mail Router Performance Counters	38

Preface

This white paper is intended to complement, rather than replace, existing resources that are specific to optimizing and maintaining the components that make up a MS Dynamics CRM implementation. To that end, this document provides relevant context for each topic, information on maintaining and optimizing that component specifically within the MS Dynamics CRM implementation, and links to related resources that may offer additional guidance for a particular environment.

Important: The optimization techniques and performance improvements provided in this paper are based on a Microsoft Dynamics CRM 4.0 implementation with a standard configuration. Depending on the level of customization in a specific Microsoft Dynamics CRM 4.0 implementation, these techniques may perform differently or yield varying results. Verify the functionality and performance impact of any of these optimization techniques before implementing them in a production environment.

In addition, for Microsoft Dynamics CRM deployments that are integrated with other systems, test optimization techniques in an environment that approximates the complexity and integration present the production environment.

Performance on test systems may not reflect that of production Microsoft Dynamics CRM servers if the test server does not have the same integrations to other systems as the production Microsoft Dynamics CRM Server. Also, performance results may vary if the database on the test system is not of similar size and structure to that in the production environment.

Caution: To protect against the risk of losing information in the system or Microsoft Windows® Active Directory® databases, create back-up copies of all databases before performing any of the optimization techniques described in this white paper.

Introduction to Optimizing and Maintaining Microsoft Dynamics CRM 4.0

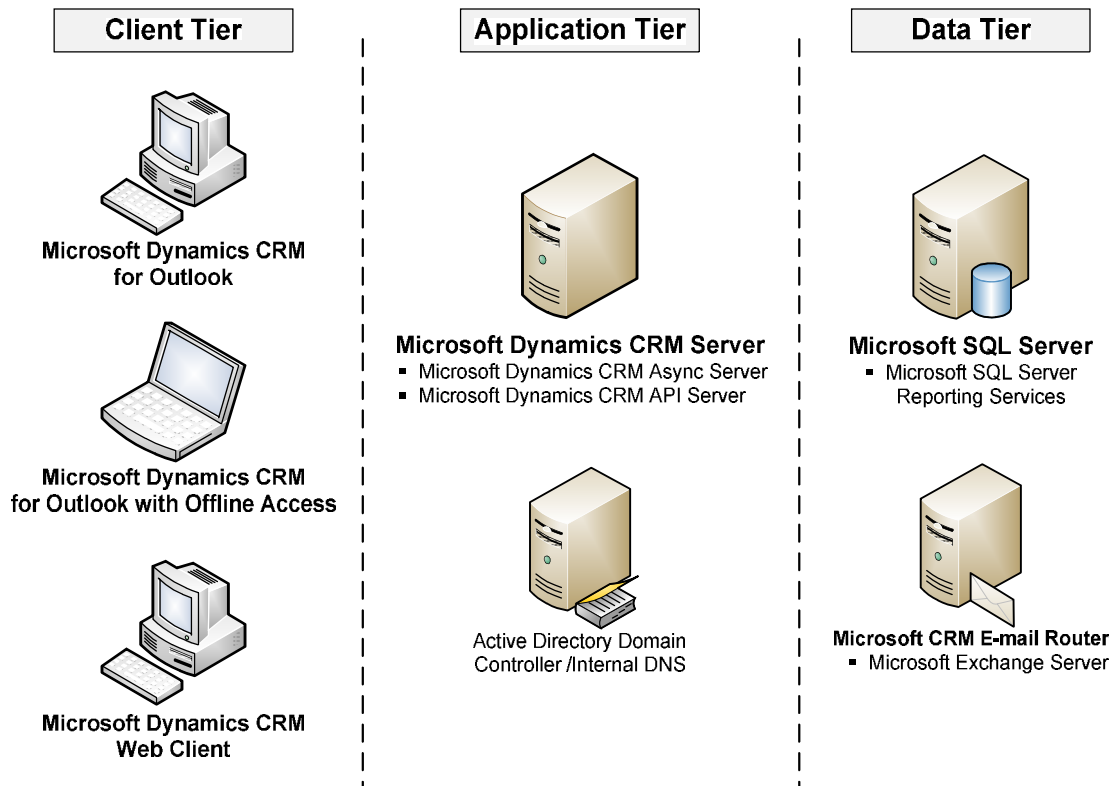
Optimizing and maintaining a connected system such as Microsoft Dynamics CRM 4.0 requires a strong knowledge of the:

- Server architecture and components that make up a specific implementation
- Potential performance issues that can adversely impact system components or their interactions
- Key techniques for identifying, troubleshooting, and resolving performance issues should they occur

Note: End-user efficiency at completing routine tasks in Microsoft Dynamics CRM 4.0 is an often overlooked consideration in assessing how well a particular implementation is addressing overall business requirements.

Server Architecture for Implementing Microsoft Dynamics CRM 4.0

While there are a variety of specific hardware and software requirements for implementing a Microsoft Dynamics CRM 4.0 solution, deployments in general have a basic server architecture in common. The following graphic depicts typical a server architecture for implementing Microsoft Dynamics CRM 4.0.



Important: This is an example of a server architecture for implementing Microsoft Dynamics CRM 4.0. Planning an implementation for your environment may indicate that an alternate structure is more appropriate for your business.

As the graphic shows, this general server architecture provides for three logical tiers within a Microsoft Dynamics CRM 4.0 implementation. Understanding the unique characteristics of each tier in the solution is a critical element in being effective at optimizing and maintaining system performance.

The Client Tier

The *client tier* provides for interactions between the Microsoft Dynamics CRM 4.0 system and its users. As a result, client tier components are configured and optimized for responsive user interaction.

The following table identifies the Microsoft Dynamics CRM 4.0 system components that typically fall within the client tier and the potential performance issues that most commonly affect the tier.

MS CRM 4.0 System Components	Potential Performance Issues
Microsoft Dynamics CRM Clients <ul style="list-style-type: none">• MS CRM for Office Outlook• MS CRM Office Outlook with Offline Access• MS CRM Web client	Very specific issues related to the network link: <ul style="list-style-type: none">• Ping time to CRM MT (latency)• Cache expiration times• Compression technology

The Application Tier

The *application tier* centralizes and performs all of the business logic processing, as well as providing the Web server responsibilities, for a Microsoft Dynamics CRM 4.0 implementation.

Note: Because of the need to balance the differing server resource requirements of the Web server components and the business processing components, configuring and tuning the application tier can be more difficult than can be optimizing the client tier.

The following table identifies the Microsoft Dynamics CRM 4.0 system components that typically fall within the application tier and the potential performance issues that most commonly affect the tier.

MS CRM 4.0 System Components	Potential Performance Issues
Microsoft Dynamics CRM Server <ul style="list-style-type: none">• MS CRM Async Server• MS CRM API server	<ul style="list-style-type: none">• Resource bottlenecks<ul style="list-style-type: none">○ Processor○ Memory• Plug-ins• SDK code

The Data Tier

The *data tier* maintains solution data and serves up that data to address requests from the application tier. The database architecture of a Microsoft Dynamics CRM 4.0 implementation includes both Microsoft SQL Server and the databases that contain the records and information unique to an organization. Depending on the implementation, the data tier can also include the Microsoft Dynamics CRM E-mail Router, as well as the e-mail server and database.

The following table identifies the Microsoft Dynamics CRM 4.0 system components that typically fall within the data tier and the potential performance issues that most commonly affect the tier.

MS CRM 4.0 System Components	Potential Performance Issues
<ul style="list-style-type: none">• Microsoft SQL Server• Microsoft SQL Server Reporting Services• Microsoft Exchange Server or Microsoft Dynamics CRM E-mail Router	<ul style="list-style-type: none">• Physical Data Layout• Indexes

Recommended Approach to Optimization and Maintenance

After initially deploying Microsoft Dynamics CRM 4.0, it is important to optimize the solution to accommodate the unique business and environmental factors inherent in a specific scenario. After the system is configured for optimal performance, be sure to establish a performance baseline by recording the optimized levels of performance.

Over the course of time, routine, daily use of a connected system such as Microsoft Dynamics CRM 4.0 implementation can adversely impact solution performance. After deploying Microsoft Dynamics CRM 4.0, be sure to leverage the functionality of the Windows Server System Monitor utility to collect and analyze performance data that you can use to monitor the overall health of the system. It is recommended to monitor performance remotely when possible, as this ensures that the load caused by the monitoring will not be reflected in the performance results observed.

Note: For general information about the guidelines and best practices for performance monitoring, see the TechNet article *Monitoring Performance*:

<http://technet2.microsoft.com/windowsserver/en/library/204bb1a3-713b-4ba3-9a9b-98ca46a3076b1033.msp?mfr=true>

Important: Microsoft Dynamics CRM 4.0 provides a number of new performance counters to assist administrators in their efforts to monitor and maintain the overall performance of Microsoft Dynamics CRM implementations. Details about these counters and their effective use will be provided with the Microsoft Dynamics CRM 4.0 Server Management Pack for Microsoft Operations Manager 2005, which is due for release in the first half of 2008.

For the latest information about Microsoft Dynamics CRM 4.0 performance and to learn about the release of additional support resources, visit the Microsoft Dynamics CRM Team blog (<http://blogs.msdn.com/crm/>).

Optimizing and Maintaining Client Tier Performance

Optimizing Network Performance

Optimizing the performance of the network that supports a Microsoft Dynamics CRM 4.0 implementation is a critical step for ensuring that the system operates efficiently.

The primary factors that define network performance are bandwidth and latency.

- *Bandwidth* is the width or capacity of a specific communications channel.
- *Latency* is the time required for a signal to travel from one point on a network to another; latency is a fixed cost between two points. To optimize for latency, eliminate as many round trips per request as possible.

Networks with high bandwidth do not guarantee low latency. For example, a network path traversing a satellite link often has high latency, even though throughput is very high. It is not uncommon for a network round trip traversing a satellite link to have five or more seconds of latency. An application designed to send a request, wait for a reply, send another request, wait for another reply, and so on, will wait at least five seconds for each packet exchange, regardless of how fast the server is.

In addition to any optimizations that apply generally to all Microsoft Dynamics CRM 4.0 implementations, there are a variety of techniques aimed specifically at overcoming latency and maximizing bandwidth utilization in an effort to ensure the best possible performance in these types of settings.

Other variables, such as the use of SSL (which does not support file caching), may make it necessary or desirable to consider alternative techniques, such as:

- Deploying Microsoft Dynamics CRM at a Regional Level
- Accessing Microsoft Dynamics CRM by using Terminal Services/Citrix

Important: It is recommended to test the implications of client performance in any WAN environment with potential bandwidth or latency issues.

Configuring Client-Side Caching

Client-side browser settings can greatly affect user experience over slower connections. Increasing the disk space reserved for temporary Internet files can help to optimize performance by ensuring adequate room for the cache to take advantage of improved load times when making subsequent visits to the same page for the same files.

► **To configure client-side browser settings:**

1. On the client computer, in Internet Explorer, on the **Tools** menu, click **Internet Options**.
1. In the **Internet Options** dialog box, under **Temporary Internet Files**, select the **Automatically check for newer versions of pages** check box, under **Amount of disk space to use**, specify a value from 200 to 300 megabytes of disk space for temporary Internet files, and then click **OK**.

Note: The first time you load Microsoft Dynamics CRM 4.0 after modifying client-side browser settings, it may take several moments longer than usual. Subsequent page loads, however, will show improved performance, as many of the Microsoft Dynamics CRM Web pages will already be cached.

Configuring Content Expiration

Content expiration controls the Web objects cache for the clients accessing CRM. By default, content expiration is configured to three days. Any implementation with fairly static content or in a WAN environment with slower connections may benefit from increasing the content expiration value to 15 days.

Making this change configures a client computer running the Microsoft Dynamics CRM Web application or the Microsoft Dynamics CRM client for Outlook to download items into the temporary Internet files without checking for newer files for a period of 15 days.

Note: To ensure the greatest performance benefit from this optimization technique, combine it with modifications to the client-side Web browser settings configuration.

► **To configure content expiration**

1. On the Microsoft Dynamics CRM Server, in the Internet Information Services (IIS) Manager administrative tool, right-click **Microsoft Dynamics CRM Web Site**, and then click **Properties**.
2. In the Properties dialog box, on the **HTTP Headers** tab, specify for content to expire after 15 days, and then click **OK**.

Note: This setting change affects client systems when the current interval expires (in less than 72 hours).

Important: Configuring the content expiration period also impacts the length of time required to complete a hotfix rollout.

Using Compression Techniques

Compression techniques designed to optimize network performance directly affect the size of the data files that are transmitted over the network. Several vendors offer web-acceleration appliances that improve the performance of applications such as CRM, and using these devices can greatly improve CRM performance over the WAN. Several Microsoft CRM customers have implemented WAN acceleration appliances with encouraging results.

Accessing Dynamics CRM by using Terminal Services/Citrix

With the high number of roundtrips that Microsoft Dynamics CRM carries out on each page, over very high latency connections, often a Terminal Services/Citrix connection will provide for improved performance.

Note: Terminal Services connections are only accessible via Microsoft Dynamics CRM for Outlook; Terminal Services does not support Microsoft Dynamics CRM for Outlook with Offline Access.

Optimizing and Maintaining Microsoft Dynamics CRM for Outlook

Hardware Requirements

Before you consider optimizing the performance of Microsoft Dynamics CRM for Outlook, verify that the client computers meet the associated hardware requirements.

Note: For the full requirements associated with running Microsoft Dynamics CRM for Microsoft Office Outlook, see the *Microsoft Dynamics CRM 4.0 Implementation Guide*: <http://www.microsoft.com/downloads/details.aspx?FamilyID=1ceb5e01-de9f-48c0-8ce2-51633ebf4714&DisplayLang=en>

The following table lists the minimum and recommended hardware requirements for Microsoft Dynamics CRM for Outlook.

Component	Minimum	Recommended
Processor	Intel Pentium III 1.0-GHz CPU, or comparable	Dual-core 1.8-GHz CPU or higher
Memory	1-GB RAM	2-GB RAM or more
Hard disk	Up to 500 MB of available hard-disk space	Up to 500 MB of available hard-disk space

Note: When connected to the network, a computer running Microsoft Dynamics CRM for Office with Offline Access has hardware requirements similar to those shown in the table. However, more powerful hardware is often required to adequately meet performance expectations when using the offline functionality provided by Microsoft Dynamics CRM for Office with Offline Access.

Optimizing the Synchronization Process

To optimize the performance of Microsoft Dynamics CRM for Outlook, configure the synchronization process so that it:

- Affects the fewest record types and occurs as infrequently as possible without compromising business requirements
- Does not create duplicate records if key fields match

► To set synchronization options for optimal performance

1. In Microsoft Dynamics CRM for Outlook, on the **CRM** menu, click **Options**.
2. In the **Set Personal Options** dialog box, on the **Synchronization** tab, under **Synchronize Microsoft Dynamics CRM items with my default folders**, select only record types that you require to be synchronized.
3. Under **Schedule automatic synchronization with Outlook**, increase the interval between each occurrence of automatic synchronization.
4. Under **Schedule what to do if duplicates are found during synchronization**, click **Do not create duplicates** to block the creation of records if key fields contain the same data.
5. Click **OK** to close the **Set Personal Options** dialog box and save the configuration.

Optimizing Address Book Performance

To optimize Address Book performance, configure settings so that the Address Book:

- Matches only against contacts synchronized to Microsoft Dynamics CRM
- Retrieves updates as infrequently as possible without compromising business requirements

► To set Address Book options for optimal performance

1. In Microsoft Dynamics CRM for Outlook, on the **CRM** menu, click **Options**.
2. In the **Set Personal Options** dialog box, on the **Address Book** tab, under **Select how e-mail recipients are reconciled with Microsoft Dynamics CRM records**, to the right of **Contacts**, verify that **Match only against contacts synchronized to Microsoft Dynamics CRM** is selected.
3. To the right of **Other record types**, ensure that **Match only the items I own** is selected, and then click **OK** to save the configuration and close the **Set Personal Options** dialog box.

Note: The Address Book tab also provide the option to configure the frequency with which the Address Book is updated from Microsoft Dynamics CRM

Bypassing the Proxy Server for On-premise Deployments within a LAN

Proxy servers often act as cache servers and improve client computer performance by helping to load web pages more quickly. However, configuring a client computer with improper proxy server settings can ultimately result in a degradation of performance.

Frequently, administrators or end-users implement automatic proxy configuration to avoid the effort required to configure settings manually. While this can provide a certain level of load balancing, depending on the complexity of the configuration script, users may experience a significant delay in accessing online resources.

When connection is established through a proxy server, the host name of the site and the proxy server name are cached. In the same session, subsequent attempts to connect to the host use the previously cached information. If the proxy server referenced in the cache is unavailable, the automatic proxy configuration script is not re-processed, and Internet Explorer displays a "Page Cannot Be Displayed" error message.

For on-premise deployments of Microsoft Dynamics CRM 4.0 within a local area network, client computers can achieve much higher throughput by completely bypassing the proxy server. The Microsoft Dynamics CRM server within the LAN offers local web addresses without requiring a proxy server. As a result, client computers can be configured to bypass a proxy server for local addresses as long as the fully qualified domain name of the Microsoft Dynamics CRM server is listed as an exception.

Note: For more information about proxy server configuration settings, view the following resources:

- *Using automatic proxy configuration*
<http://technet2.microsoft.com/windowsserver/en/library/1f2bad68-8caf-4613-bf4d-c4f6a7d6acf81033.msp?mfr=true>
- *Take the Burden Off Users with Automatic Configuration in .NET*
<http://msdn.microsoft.com/msdnmag/issues/05/08/AutomaticProxyDetection/default.aspx>
- *How to disable automatic proxy caching in Internet Explorer*
<http://support.microsoft.com/kb/271361>
- *You experience a delay when you use your Windows XP computer to log on to a domain or to connect to a network resource*
<http://support.microsoft.com/kb/832161>

Improving Startup Time for Microsoft Dynamics CRM for Outlook

Microsoft Dynamics CRM for Outlook relies on the Microsoft Dynamics CRM Application Host system tray application to maintain a connection to Microsoft Dynamics CRM. Configuring Microsoft Dynamics CRM for Outlook so that the Host service runs continuously, rather than only when Microsoft Office Outlook runs, will help to ensure that Microsoft Dynamics CRM for Outlook starts and loads as efficiently as possible.

► To ensure that the Microsoft Dynamics CRM Host service runs continuously

1. In Microsoft Dynamics CRM for Outlook, on the **CRM** menu, click **Options**.
2. In the **Set Personal Options** dialog box, on the **General** tab, under **Improve Outlook startup time and Microsoft Dynamics CRM display speed**, ensure that **Always run the Microsoft Dynamics CRM Application Host process** is selected, and then click **OK** to close the **Set Personal Options** dialog box.

Disabling Extraneous Processes and Applications

All the processes and applications running on a client computer can adversely affect the performance of the Microsoft Dynamics CRM Client for Outlook, especially for client computers configured for offline access. Depending on the required level of client performance, consider turning off non-critical business processes, third-party add-ins for Microsoft Office Outlook, and software, such as games and music programs.

Reducing Exchange Traffic by Modifying Outlook Rules

For deployments that include Microsoft Exchange Server and the Microsoft Dynamics CRM E-mail router configured to use a forward mailbox strategy, modifying the Outlook forwarding rules can reduce the volume of Exchange traffic on the system. For users who track only CRM e-mails, configure Outlook to forward only those messages that include **CRM:** is in the subject or body.

Note: **CRM:** is the default prefix specified for the tracking token in the System Settings > E-mail tracking tab in CRM settings. If you've changed your prefix, change the instructions to use the value in the prefix setting in System Settings.

This configuration can reduce the volume of traffic being forwarded around the Exchange server, as well as limiting the risk that the sink mailbox will overflow should the Exchange Router service quit functioning.

► **To modify the Outlook e-mail forwarding rule:**

1. In Microsoft Office Outlook, on the **Tools** menu, click **Rules and Alerts**.
2. In the **Rules and Alerts** dialog box, on the **E-mail Rules** tab, select the existing CRM forwarding rule, and then in the **Change Rule** drop-down list, click **Edit Rule Settings**.
3. In the Rules Wizard, under **Step 1: Select condition(s)**, check the option box to the left of **with specific words in the subject or body**.
4. Under **Step 2: Edit the rule description**, specify **CRM:**, and then in the Rules Wizard, click **Finish** to save the rule.

Note: Additional information about optimizing and maintaining the Microsoft Dynamics CRM E-mail Router is available in the *Optimizing the Data Tier* section of this white paper. For information about optimizing and maintaining Microsoft Exchange Server, in the Exchange Server TechCenter, view the following resources:

- *Performance and Scalability Guide for Exchange Server 2003*
[http://technet.microsoft.com/en-us/library/aa996078\(EXCHG.65\).aspx](http://technet.microsoft.com/en-us/library/aa996078(EXCHG.65).aspx)
- *Monitoring Exchange 2007 with Microsoft Operations Manager 2005 SP1*
[http://technet.microsoft.com/en-us/library/bb201735\(EXCHG.80\).aspx](http://technet.microsoft.com/en-us/library/bb201735(EXCHG.80).aspx)

Optimizing Offline Synchronization

When optimizing offline synchronization of Microsoft Dynamics CRM for Outlook with Offline Access, consider the following best practices.

- Install high-speed connections at remote branches to improve synchronization performance for offline clients.
- Assign all users roles that are defined with the *minimum* access levels and permissions required to perform a job function to help to guarantee optimized data synchronization to the offline client.
- Whenever possible, avoid using:
 - "Parent downloaded=true" clauses. Use of this clause often results in the synchronization of unnecessary data, which can degrade the performance of the synchronization process.
 - *Sliding time windows*. A variable that calculates results from date range that shifts depending on the current date, for example "last 60 days," can adversely affect performance. Instead, use variables that calculate results from a specific date forward.
- Implement local data filtering for each offline client to ensure that users have offline access only to the data required to perform their job functions. After implementing local data filtering, be sure to remain online and synchronize the data manually. The initial synchronization will be slower than subsequent synchronizations because Microsoft Dynamics CRM must remove records
 - ▶ **To deactivate unused local data groups**
 1. In Microsoft Dynamics CRM for Outlook, on the **CRM** menu, click **Modify Local Data Groups**.
 2. In the **Local Data** dialog box, on the **Data Groups** tab, select any unused data groups, and then use the toolbar to deactivate the selected data groups.
Note: Deactivated groups appear on the **Inactive Data Groups** tab.
 - Configuring offline synchronization to run in the background periodically, for example every 15 minutes, can improve performance by helping to ensure that:
 - The user's local database is regularly updated so that when the user actually wants to go offline, there's not as much data to copy down
 - If the user loses connection and the client goes into an offline mode, the data store is relatively current
 - ▶ **To schedule Offline Synchronization to run in the background**
 1. In Microsoft Dynamics CRM for Outlook, on the **CRM** menu, click **Modify Local Data Groups**.
 2. In the **Local Data** dialog box, on the **Data Groups** tab, select any unused data groups, and then use the toolbar to deactivate the selected data groups.
Note: Deactivated groups appear on the **Inactive Data Groups** tab.

Note: For more information about optimizing the offline synchronization process, view the following resources:

- *Speed up data synchronization*
<http://www.microsoft.com/dynamics/crm/using/configure/improvesync.aspx>
- *CRM Data - Microsoft Dynamics CRM Laptop Client*
<http://www.microsoft.com/dynamics/crm/using/sales/localdatagroup.aspx>

- *Hitting the Road with Microsoft Dynamics CRM for Outlook with Offline Access*
http://rc.crm.dynamics.com/rc/regcont/en_us/OP/articles/localdatagroup.aspx

Microsoft Dynamics CRM for Outlook Diagnostics

It is recommended to run Microsoft Dynamics CRM for Microsoft Office Outlook Diagnostics as part of a regular maintenance plan for Microsoft Dynamics CRM for Outlook. This tool performs a variety of diagnostic tests to check for issues that may affect performance, connectivity, configuration, and synchronization.

Note: If the tool encounters issues as runs, common issues, such as when the Microsoft Dynamics CRM Add-in is disabled, will be addressed automatically. The tool will help users troubleshoot problems, such as when synchronization error occurs because a user has typed invalid characters. It may also refer to a knowledge-based article to help in the resolution of other problems.

The diagnostics tool can be started in Support Mode, which is designed to provide system administrators with quick access to relevant information from the CRM add-in, synchronization, and system environment. In Support Mode, administrators can use on/off toggles to specify which tests will run and which files will be generated to evaluate errors.

Important: Carefully consider the performance implications of enabling various diagnostic options. For example, when aggressive e-mail tagging is enabled, each received e-mail will result in e-mail tagging cycle processing, and when e-mail background promote is enabled, each received e-mail will be processed to see if it should be tracked in CRM.

Note: For more information about Microsoft Dynamics CRM for Microsoft Outlook Diagnostics, see the *Microsoft Dynamics CRM 4.0 Implementation Guide*:
<http://www.microsoft.com/downloads/details.aspx?FamilyId=1CEB5E01-DE9F-48C0-8CE2-51633EBF4714&displaylang=en>

Optimizing and Maintaining Application Tier Performance

Optimizing and Maintaining Microsoft Windows Server

Optimizing the Performance of Microsoft Windows Server

In many respects, Windows Server is foundational component in a Microsoft Dynamics CRM 4.0 implementation. As a result, proper configuration of the Windows Server operating system is a critical step in ensuring the optimal performance of any Microsoft Dynamics CRM 4.0 deployment.

Windows Server Configuration Options

Windows Server includes a variety of settings that you can configure to incrementally improve performance. These settings can be grouped according to the target area for performance improvement. For example, you can use specific settings to optimize Windows Server 2003 for networking, storage, IIS, file servers, or Active Directory.

Note: For more information about configuring Windows Server 2003 settings for optimal performance, view the following resources:

- *Performance and Scalability*
<http://www.microsoft.com/windowsserver2003/evaluation/performance/default.mspx>
- *Windows Server 2003 Solution Center*
<http://support.microsoft.com/?scid=ph;en-us;3198>

Increasing the Ephemeral TCP Port Limit

In some situations, you may want to reserve a range of ports so that a program or process that requests a random port will not be assigned a port that is in the reserved range. When you reserve a range of ports, only a program or process that specifically requests a port that is in the reserved range can use the port. By default, Windows Server 2003 is configured to allow a maximum of 5,000 ephemeral TCP ports.

► To increase the maximum number of ephemeral TCP ports

1. In the Registry Editor, navigate to the subkey:
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters
2. Click **Parameters**, and then, on the **Edit** menu, click **New**.
3. Create a registry entry using the following information:

<u>Value Name</u>	MaxUserPort
<u>Value Type</u>	DWORD
<u>Value data</u>	65534
<u>Valid Range</u>	5000-65534 (decimal)
<u>Default</u>	0x1388 (5000 decimal)
<u>Description</u>	Controls the maximum port number that is used when a program requests any available user port from the system. Typically, ephemeral (short-lived) ports are allocated between the values of 1024 and 5000, inclusive.

4. Close the Registry Editor, and then restart the computer to apply the new configuration.

You can also use the 'TCPTimedWaitDelay' registry parameter to specify the period before a closed port becomes reusable.

Note: For more information, see the article *When you try to connect from TCP ports greater than 5000 you receive the error 'WSAENOBUFFS (10055)'*:

<http://support.microsoft.com/default.aspx/kb/196271>

Monitoring the Performance of Microsoft Windows Server

You can use Microsoft Windows Server performance counters to monitor the operating system. Windows Server provides performance counters that you can use to help identify potential performance bottlenecks associated with memory and the cache, processors (and multi-processor computers), physical disks, and the network infrastructure. Remember to monitor each disk and processor used by the operating system.

Note: For more information about monitoring the performance of Microsoft Windows Server, view the following resources:

- *Monitoring Performance*
<http://technet2.microsoft.com/windowsserver/en/library/8e9e5b72-13af-4525-9b60-52864b9d49b21033.msp?mfr=true>
- *Windows Server "Longhorn" Performance and Reliability Monitoring Step-by-Step Guide*
<http://technet2.microsoft.com/windowsserver2008/en/library/7e17a3be-f24e-4fdd-9e38-a88e2c8fb4d81033.msp?mfr=true>

Optimizing and Maintaining the Microsoft .NET Framework and Microsoft .NET Applications

Optimizing the Performance of the Microsoft .NET Framework

Configuring the .NET Framework for optimal performance involves tuning the common language runtime (CLR) and then, depending on the nature of any specific application, tuning the associated .NET Framework technology, for example ASP.NET-connected applications, Web services, Enterprise Services, and ADO.NET code.

Potential Performance Issues

When tuning the CLR to optimize the Microsoft .Net Framework, be sure to consider the potential issues described in the following table:

Potential Issue	Description
Memory misuse	Creating too many objects, or failing to properly release resources, pre-allocate memory, or explicitly force garbage collection can prevent the CLR from efficiently managing memory, which can lead to an increased working set size.
Resource cleanup	Implementing finalizers unnecessarily, failing to suppress finalization in the Dispose method, or failing to release unmanaged resources can lead to unnecessary delays in reclaiming resources and can potentially create resource leaks.
Improper use of threads	Creating threads on a per-request basis and not sharing threads using thread pools can cause performance and scalability bottlenecks for server applications.
Abusing shared resources	Creating resources per request can lead to resource pressure, and failing to properly release shared resources can cause delays in reclaiming them.
Type conversions	Implicit type conversions and mixing value and reference types leads to excessive boxing and unboxing operations.
Misuse of collections	Each collection type in the .NET Framework class library is designed to meet specific storage and access requirements; they may not perform optimally outside of those requirements.

Potential Issue	Description
Inefficient loops	Looping magnifies even the slightest coding inefficiency, and loops that access an object's properties are a common culprit of performance bottlenecks, particularly if the object is remote or the property getter performs significant work.

Note: For more information about optimizing code for efficient CLR processing, in the MSDN prescriptive guidance *Improving .NET Application Performance and Scalability*, view the following resources:

- *Chapter 5 – Improving Managed Code Performance*
<http://msdn2.microsoft.com/en-us/library/ms998547.aspx>
- *Checklist: Managed Code Performance*
<http://msdn2.microsoft.com/en-us/library/ms979052.aspx>

Monitoring the Performance of Microsoft .NET Framework Applications

The following table describes potential bottlenecks that can occur in applications written using managed code and how to identify those bottlenecks by using system counters.

Bottleneck	Description
Excessive memory consumption	Excessive memory consumption can result from poor managed or unmanaged memory management. To identify this symptom, watch the following performance counters: <ul style="list-style-type: none"> ▪ Process\Private Bytes ▪ .NET CLR Memory\# Bytes in all Heaps ▪ Process\Working Set ▪ .NET CLR Memory\Large Object Heap size An increase in 'Private Bytes' when the '# of Bytes in all Heaps' counter remains the same indicates unmanaged memory consumption. An increase in both counters indicates managed memory consumption.
Large working set size	The working set is the set of memory pages currently loaded in RAM. This is measured by 'Process\Working Set'. A high value might indicate that you have loaded several assemblies. Unlike other counters, 'Process\Working Set' has no specific threshold value to watch, although a high or fluctuating value can indicate a memory shortage. A high or fluctuating value accompanied by a high rate of page faults clearly indicates that the server has insufficient memory.
Fragmented large object heap	Objects larger than 83 KB are allocated in the large object heap, which is measured by '.NET CLR Memory\Large Object Heap size'. Frequently, these objects are buffers (large strings, byte arrays, and so on) used for I/O operations (for example, creating a 'BinaryReader' to read an uploaded image). Such allocations can fragment the large object heap; consider recycling those buffers to avoid fragmentation.
High CPU usage	High CPU usage is usually caused by poorly written managed code, such as code that: <ul style="list-style-type: none"> ▪ Causes excessive garbage collection (measured by '% Time in GC') ▪ Throws many exceptions (measured by '.NET CLR Exceptions\# of Exceps Thrown /sec.') ▪ Creates many threads, causing the CPU to spend large amounts of time switching between threads instead of performing real work (measured by 'Thread\Context Switches/sec')
Thread contention	Thread contention occurs when multiple threads try to access a shared resource. To identify thread contention, monitor: <ul style="list-style-type: none"> ▪ .NET CLR LocksAndThreads\Contention Rate / sec

Bottleneck	Description
	<ul style="list-style-type: none"><li data-bbox="459 268 1372 300">▪ .NET CLR LocksAndThreads\Total # of Contentions To reduce the contention rate, identify and fix the code that accesses shared resources or uses synchronization mechanisms.

Note: For more information about monitoring the performance of Microsoft .NET framework applications, view the following resources:

- *Performance Counters in the .NET Framework*
<http://msdn2.microsoft.com/en-us/library/w8f5kw2e.aspx>
- *Monitoring ASP.NET Performance*
<http://technet2.microsoft.com/windowsserver/en/library/0b792d7c-30ca-4349-af4c-e000643aa6f31033.msp?mfr=true>

Optimizing and Maintaining Internet Information Services

Optimizing the Performance of Internet Information Services

Optimizing the performance of Microsoft Internet Information Services (IIS) within a Microsoft Dynamics CRM 4.0 implementation benefits not only the overall system, but also any custom applications, plug-ins, or add-ins that have been developed by using the *Microsoft Dynamics CRM 4.0 SDK*.

Note: For additional information about optimizing the performance of IIS, view the following resources:

- *Performance Tuning (IIS 6.0):*
<http://www.microsoft.com/technet/prodtechnol/WindowsServer2003/Library/IIS/71490aae-f444-443c-8b2a-520c2961408e.msp?mfr=true>
- *IIS 6.0 Operations Guide (IIS 6.0)*
<http://www.microsoft.com/technet/prodtechnol/WindowsServer2003/Library/IIS/dc9cb2-9270-4ea5-8556-46528fea058d.msp?mfr=true>

Optimization Considerations

When optimizing the performance of IIS on the computer running Microsoft Dynamics CRM 4.0 Server, be sure keep in mind the following considerations.

- Tracing and debugging are disabled by default and may cause performance issues if enabled. If necessary disable tracing and debugging by configuring the Machine.config and Web.config files as shown in the following sample:

```
<configuration>
  <system.web>
    <trace enabled="false" pageOutput="false" />
    <compilation debug="false" />
  </system.web>
</configuration>
```

Optimizing Microsoft .NET ThreadPool Settings

You can modify parameters in the Machine.config file to accommodate a specific environment. However, if each .aspx page makes a Web service call to a single IP address, it is recommended to adjust these parameters as shown in the following table:

Parameter	Value
maxWorkerThreads	100
maxIoThreads	100
maxconnection	12*n (where <i>n</i> is the number of CPUs)
minFreeThreads	88*n
minLocalRequestFreeThreads	76*n
minWorkerThreads	50 (manually add this parameter and value to the file)

Note: For more information about configuring Microsoft .NET ThreadPool settings, see the Knowledge Base article *Contention, poor performance, and deadlocks when you make Web service requests from ASP.NET applications*:

<http://support.microsoft.com/kb/821268>

Several of these recommendations include a formula to calculate the number of CPUs on a server. The variable that represents the number of CPUs in the formulas is N . For these settings, if you have hyperthreading enabled, you must use the number of logical CPUs instead of the number of physical CPUs. For example, if you have a four-processor server for which hyperthreading has been enabled, the value of N in the formulas will be **8** instead of **4**.

Note: When you use this configuration, you can execute a maximum of 12 ASP.NET requests per CPU at the same time because $100 - 88 = 12$. Therefore, at least $88 * N$ worker threads and $88 * N$ completion port threads are available for other uses (such as Web service callbacks).

Using the example of a server with four processors and hyperthreading enabled, these formulas result in the following configuration:

```
<processModel maxworkerThreads="100" maxIoThreads="100"
minworkerThreads="50"><httpRuntime minFreeThreads="704"
minLocalRequestFreeThreads="608"><connectionManagement><add
address="[ProvideIPHere]" maxconnection="96"/></connectionManagement>
```

Note: For more information, in the MSDN prescriptive guidance *Improving .NET Application Performance and Scalability*, see Chapter 6 – *Improving ASP.NET Performance*:

<http://msdn2.microsoft.com/en-us/library/ms998549.aspx>

For general information about how to configure IIS to improve performance of Web service calls from ASPX pages, see the MSDN article *At Your Service: Performance Considerations for Making Web Service Calls from ASPX Pages*:

<http://msdn2.microsoft.com/en-us/library/Aa480507.aspx>

Monitoring the Performance of Internet Information Services

Microsoft Dynamics CRM server is basically an Internet Information Services (IIS) server that runs a Microsoft.NET-connected application.

To monitor the overall health of the servers, you should collect information about several Windows Server 2003 counters. One of the key counters to monitor and measure against a baseline is the %Process Time for the inetinfo (IIS). If the Microsoft Dynamics CRM server meets the recommended hardware requirements and does not perform any other tasks, the server should not experience any associated performance issues.

Note: For more information about monitoring the performance of Internet Information Services, view the following resources:

- *IIS 6.0 Performance Counters*
<http://technet2.microsoft.com/windowsserver/en/library/0e6a0151-bf62-4d52-89ad-59cc0ae6da4c1033.msp?mfr=true>
- *How to monitor Web server performance by using counter logs in System Monitor in IIS*
<http://support.microsoft.com/default.aspx?scid=kb%3BEN-US%3Bq313064>

Optimizing the Performance of Microsoft Dynamics CRM 4.0

To configure Microsoft Dynamics CRM 4.0 for optimal performance, focus attention on both the MS CRM Server as well as any MS Dynamics CRM 4.0 applications.

Installing Microsoft Dynamics CRM on Multiple Servers

Microsoft Dynamics CRM 4.0 supports load balancing across multiple servers. You can use multiple CRM 4.0 installations to balance the processing load across several servers.

With multiple servers, you can also implement departmental Microsoft CRM systems that still have access to the same Microsoft CRM database.

Note: For more information about how to implement Microsoft Dynamics CRM 4.0 on multiple servers, see "Multiple-Server Deployment" in the *Microsoft Dynamics 4.0 Planning Guide*:

<http://www.microsoft.com/downloads/details.aspx?FamilyId=1CEB5E01-DE9F-48C0-8CE2-51633EBF4714&displaylang=en>

Implementing Server Roles

In Microsoft Dynamics CRM 4.0 Enterprise Edition, you can improve performance by installing certain Microsoft Dynamics CRM Server components and services, representing a *server role*, on different computers. By default, Setup installs all Microsoft Dynamics CRM server roles on the local computer. However, if you are installing Microsoft Dynamics CRM 4.0 Enterprise Edition, you can specify to deploy one or both of two *server role groupings*, on that computer, or to have two or more computers running the same role, which can provide load balancing benefits

The following table describes the server role groupings provided in Microsoft Dynamics CRM 4.0 Enterprise Edition.

Server Role Grouping	Description
Application Server	Provides the Microsoft Dynamics CRM 4.0 Web user interface and services
Platform Server	Enables deployment of asynchronous services, such as the Workflow and Bulk E-mail services, to a separate computer

Note: For more information about recommendations for isolating server roles or running multiple computers with the same server role, see the *Advanced Planning Deployment* chapter in the *Microsoft Dynamics CRM 4.0 Planning Guide*:

<http://www.microsoft.com/downloads/details.aspx?FamilyId=1CEB5E01-DE9F-48C0-8CE2-51633EBF4714&displaylang=en>

Disabling Microsoft Dynamics CRM 4.0 Server Platform Tracing

Only enable Microsoft Dynamics CRM 4.0 Server platform tracing to assist in troubleshooting efforts, and disable it immediately upon resolving any issues.

► To disable Microsoft Dynamics CRM 4.0 Server platform tracing

1. In the Registry Editor, navigate to
HKEY_LOCAL_MACHINE | Software | Microsoft | MSCRM

With Server platform tracing enabled, the registry includes the keys:

- TraceDirectory
- TraceCategories
- TraceRefresh
- TraceCallStack
- TraceSchedule
- TraceEnabled

2. If necessary, set the value of the 'TraceEnabled' registry key to 0, increase the value of the 'TraceRefresh' registry key by one, and then save the key.

Optimizing the Performance of Microsoft Dynamics CRM 4.0 Applications

Configuring Microsoft Dynamics CRM 4.0 applications for optimal performance requires attention to the following areas:

- The Microsoft Dynamics CRM Web application

- Microsoft Dynamics CRM customizations
- Custom Microsoft Dynamics CRM SDK applications

Optimizing the Performance of the Microsoft Dynamics CRM Web Application

You can improve the performance of the Web application with some basic changes to the initial configuration of Microsoft Dynamics CRM.

Modifying the Default View

Viewing all records or accounts every time you start Microsoft Dynamics CRM can be resource intensive, especially as the size of the database increases. To improve overall system performance, use the **Settings** page to access and customize default views to limit the records or accounts that are displayed. For example, you can configure the default view to display only the active accounts owned a user, rather than displaying all the active accounts for the entire organization.

► **To modify the default view for Accounts to display active accounts to users**

1. In Microsoft Dynamics CRM, on the **Home** page, on the **Settings** tab, click **System Customization**.
2. Under **Customize Entities**, double-click **Account**, and then under **Entity: Account**, click **Forms and Views**.
The Default view is marked with a star and lists its type as **Default Public View**.
3. Select the default view (such as **My Active Accounts**)
4. On the action toolbar, click **More Actions**, and then select **Set Default**.
5. On the **File** menu, click **Save**, and then on the **Actions** menu, click **Publish**.

Modifying Quick Find Search Columns

Quick find views search a predefined list of fields, and the number of fields included in the list can have a direct impact on performance. To ensure optimal performance, use Quick Find feature to search only on the fields required to address business needs. In addition, whenever possible, search on fields that exist in a SQL Server index.

► **To view and modify the columns searched by default**

1. In Microsoft Dynamics CRM, on the **Home** page, on the **Settings** tab, click **System Customization**.
2. Under **Customize Entities**, double-click **Account**, and then under **Entity: Account**, click **Forms and Views**.
3. Select the **Quick Find Active Accounts** view, and then under **Common Tasks**, click **Add Find Columns**.
The **Add Find Columns** dialog box displays all fields available for the Account entity; search will include any fields selected in the list. For example, if **Account Name** and **Account Number** are selected, the Quick Find feature searches both columns for the specified text.
4. In the **Quick find view: Accounts** form, select check boxes associated with the columns that you want to include in Quick Find search and clear check boxes for columns you do not want to include.
5. On the **File** menu, click **Save**, and then on the **Actions** menu, click **Publish**.

Important: To minimize the performance impact on the Microsoft SQL Server that is used for Microsoft Dynamics CRM, whenever possible, search on three or fewer columns and select indexed columns, such as **Address1: Street 1**, that are appropriate for creating a SQL index. You can also improve performance by adding a non-clustered index, such as **Address1:ZIP/Postal Code**, to search columns.

Caution: When performing Quick Find queries on entities that include the **Parent Customer** field, searching on that field requires a table scan of both the Account and Contact tables. This can increasingly affect overall system performance as the number of Account and Contact records in the associated database grows.

Optimizing the Performance of Microsoft Dynamics CRM Customizations

When optimizing the performance of Microsoft Dynamics CRM customizations, keep in mind the following guidelines:

- *Carefully consider the potential effects on your organization's business before removing or changing a column in the ORDER BY clause.* There may be important business reasons associated with the order of display in query results. Diminishing the business value of a query to improve its performance is not a viable tradeoff.
- *Use an iterative process to determine which index best optimizes query performance.* Test each index using a variety of selection criteria that may be common for the specific query. While one set of criteria may yield the expected performance increase from an index, different criteria may have no effect.

In addition, be sure to consider the following optimization techniques.

Disabling Auto-Complete on Lookups

Microsoft Dynamics CRM 4.0 introduced a new feature that provides auto-complete functionality on lookup fields. Enabled by default, this feature can help to increase user efficiency. However, auto-complete can also impact overall performance and resource utilization in a Microsoft Dynamics CRM 4.0 solution. To optimize an implementation that is not meeting performance requirements, where possible, it is recommended to disable this functionality.

Note: In certain scenarios, turning off the Forms Assistant can also help to improve the performance of a Microsoft Dynamics CRM 4.0 implementation.

Optimizing Custom Entity Creation and Importation

Turning off activities or notes when they are not required will improve the efficiency of the system at creating and importing custom entities.

Querying on Custom Entities

When you add columns to Microsoft Dynamics CRM database entities, the columns are included in an extension table rather than the entity's base table. To optimize the performance of queries on custom entities, ensure that all columns on the *ORDER BY* clause derive from a single table, and build an index that satisfies the *ORDER BY* requirements and as much of the query's *WHERE* clause selection criteria as possible. Determining this ideal index is likely to be an iterative process. However, when you implement this correctly, the performance benefit can be very significant.

Optimizing the Performance of Custom Microsoft Dynamics CRM SDK Applications

It is also important ensure the optimal performance of any custom applications, plug-ins, or add-ins that have been developed by using the Microsoft Dynamics CRM 4.0 SDK.

A specific recommendation for any custom application is to limit any columns and rows retrieved to those required to achieve the application's business goals. This technique is especially important when Microsoft Dynamics CRM users access the data from a Wide Area Network (WAN) with higher network latencies. You can limit the data returned by custom applications by using *Condition* attributes to restrict the data that the FetchXML and ConditionExpressions queries return, and by using paging to restrict the number of rows returned by a custom application.

Important: For Microsoft Dynamics CRM deployments that are integrated with other systems, test custom applications in an environment approximating the complexity and integration present the production environment. Also, performance results may vary if the database on the test system is not of similar size and structure to that in the production environment.

Optimizing and Maintaining Microsoft Dynamics CRM Reporting Services

Report server performance can be affected by a combination of factors including hardware, number of concurrent users accessing reports, the amount of data in a report, and output format.

Organizations with smaller data sets and fewer users can use either a single-server deployment, or a multiple-server deployment with one computer that is running Microsoft SQL Server for Microsoft Dynamics CRM, and another computer for running Reporting Services. With larger datasets or more users, performance will decrease quickly when complex reports are run.

When considering how to optimize Microsoft Dynamics CRM 4.0 Reporting Services, keep in mind the following guidelines:

- Report processing and rendering are memory intensive operations, so ensure that the computer hosting the report server should includes ample memory.
- Host the report server and the report server database on separate computers rather than hosting both on a single high-end computer.
- If all reports are processing slowly, consider a scale-out deployment where multiple report server instances support a single report server database. For best results, use load balancing software to distribute requests evenly across the deployment.
- If a single report is processing slowly, tune the query if the report must run on demand. You might also consider caching the report or running it as a snapshot.
- If all reports process slowly in a specific format (for example, while rendering to PDF), consider file share delivery, adding more memory, or using another format.

Note: For information about tools and guidelines for configuring Microsoft Dynamics CRM Reporting Services for optimal performance, view the following resources:

- *Planning for Scalability and Performance with Reporting Services*
<http://go.microsoft.com/fwlink/?LinkID=70650>
- *Using Visual Studio 2005 to Perform Load Testing on a SQL Server 2005 Reporting Services Report Server*
<http://msdn2.microsoft.com/en-us/library/aa964139.aspx>

For information about monitoring the performance of the Report Server, see *Monitoring Report Server Performance*:

<http://msdn2.microsoft.com/en-us/library/ms159809.aspx>

For information about mitigating performance issues by tuning memory management configuration settings, see *Configuring Available Memory for Report Server Applications*:

[http://msdn2.microsoft.com/en-us/library/ms159206\(SQL.100\).aspx](http://msdn2.microsoft.com/en-us/library/ms159206(SQL.100).aspx)

Optimizing Report Performance

Optimization Guidelines

End users want reports to come up quickly, with as little impact as possible on other users of the Microsoft Dynamics CRM database. As a result, the design and deployment of reporting functionality can be a major factor in overall report performance.

To help ensure optimal report performance, keep in mind the following guidelines:

- Configure reports to display information that occurred during a specified time frame, for example the previous 90 days, rather than to display all records in the Microsoft Dynamics CRM database.
- Reports with a large dataset or a complex SQL query should not be available on-demand to all users. Instead, schedule a snapshot in Report Manager during a time schedule when the system is lightly loaded.
- Deploy reports through Microsoft CRM, and then use Report Manager to run the reports at a scheduled time and have the results posted.
- Reports should access the fewest datasets possible to meet business requirements.

Optimization Techniques

Consider the following techniques to help ensure that reports perform optimally and minimize the potential impact of reporting on the rest of the system.

Use SQL 'Group By'

Use SQL 'Group By' to ensure that summary level data is gathered directly rather than by retrieving thousands of records and then post aggregating in reporting services. This helps to prevent the computer running Microsoft SQL Server from being hammered in gathering, transmitting then processing large volumes of data. Instead, it uses the natural indexing and grouping ability of SQL Server to massively reduce this overhead.

Create Custom Attributes

Provide custom attributes to ensure that all the reporting data is available on the [CustomEntity]ExtensionBase table, rather than on a combination of the [CustomEntity]Base and {CustomEntity}ExtensionBase tables. This avoids a join, avoiding both processing and additional temp db use. This also incurs overhead in the entity callouts to copy small amounts of data to shadow attributes in the extensionbase table (using the sdk) to enable single table reporting

Making Reports Pre-Filterable

When you create a report, you can make it "pre-filterable" by configuring it with a default filter that users can edit before they run the report. For each user that customizes and runs a pre-filterable report, the result is to effectively reduce the size of the data set and limit the amount of data pulled.

A key advantage of making a report pre-filterable is that the default filter prevents users from unintentionally running the report on all records. By default, this filter selects active records that were modified in the last 30 days. If you have the Manage Reports privilege, you can define specific default criteria for the default filter for each report. An advantage for users is the ability to edit the filter so that it locates the specific data that they require on the initial run.

For Microsoft Dynamics CRM to make a report pre-filterable, you must specify the CRMAF_ prefix in your SQL query when you create your report in Report Designer. When you add this prefix to at least one filtered view in the query, Microsoft Dynamics CRM adds a default filter to the report. For each filtered view that has this prefix in the query, users can edit filter criteria. For example, if your query includes the FilteredAccount and FilteredContact views, and your SQL query uses CRMAF_FilteredAccount and FilteredContact, the report will have a default filter. Users will be able to edit criteria related to accounts, but will be unable to edit criteria related to contacts.

Using Dynamic Excel or Filtered View Queries

To limit the number of records a report returns if you are using a dynamic Excel worksheet or using a Filtered View query (this includes FilteredView queries in custom Microsoft Dynamics CRM SQL Reporting Services reports), consider making it more restrictive. If a field in the WHERE clause is used frequently, verify that a non-clustered index exists on that field.

Optimizing Workflow

Enabling new workflow options in Microsoft Dynamics CRM 4.0 can affect the overall performance of the implementation. When considering how to ensure that Microsoft Dynamics CRM workflow functionality performs optimally for a specific implementation, keep in mind the following best practices.

- *Define the business purposes for implementing workflow prior to enabling the functionality.* During planning, analyze the business scenario and determine the primary goals of workflow within the solution. Microsoft Dynamics CRM 4.0 workflow functionality can provide for businesses process automation, exception handling, and end-user alerts.
- *Determine the appropriate security/permissions model for workflow.* With clearly established business goals in place, determine the scope of users that will be affected by the workflow implementation. Be sure to identify the users who will create and maintain workflows; apply and track workflows; and troubleshooting workflow issues.
- *Use the Scope property judiciously.* The Scope property associated with workflow rules defines the breadth of records affected by that rule. For example, rules configured with a User scope affect only the records owned by that user, while rules configured with an Organization scope will affect all records within organization, regardless of which user owns each record. Be sure to specify the appropriate scope value for each workflow rule to minimize the number of related system events.
- *Review workflow logic carefully.* Workflows that include infinite loopbacks, which because of semantic or logic errors, can never terminate through normal means, can greatly impact overall workflow performance. When creating and implementing workflow functionality within a Microsoft Dynamics CRM 4.0 deployment, be sure to

review the logic in workflow rules and any associated plug-ins for potential loopback issues. Additionally, as part of ongoing maintenance efforts, periodically publish workflow rules and review them to ensure duplicated workflow rules are not affecting the same records.

-
- *Use caution when defining workflows that are triggered on update events.* Given the frequency at which Update events occur, be precise in specifying which attributes the system “listens to” to trigger updates. In addition, avoid using “wait” states in workflows that are triggered on Update events
 - *Monitor the Microsoft Dynamics CRM 4.0 database for excess workflow log records.* Over time, the number of workflow records in the Microsoft Dynamics CRM 4.0 database will grow, which can lead to decreased performance. To limit any potential impact on performance, periodically remove excess workflow log records from the Microsoft Dynamics CRM database.
 - *To improve performance in large deployments, scale out as necessary.* For large-scale deployments, use dedicated machines to run the Async service. In addition, be sure to provide a dedicated API Server for each instance of the Async service.
 - *Test workflows.* Be sure to test and monitor the performance of new workflow functionality before implementing it in a production environment.

Important: If the existing workflow functionality within a Microsoft Dynamics CRM 4.0 implementation fails to perform properly, verify that the Async service is running properly. Often, restarting the Async service will restore Workflow functionality.

Optimizing and Maintaining Data Tier Performance

Optimizing and Maintaining Microsoft SQL Server and the Microsoft Dynamics CRM Database

Microsoft SQL Server performance can be affected by a variety of factors ranging from poor database design to improper server configuration. In addition to performing the standard recommended practices for maintaining Microsoft SQL Server, remember to implement an alert strategy to send notifications if the computer reaches any of a pre-determined set of performance thresholds.

Note: For more information about maintaining the performance of Microsoft SQL Server, view the following resources:

- *SQL Server Performance Dashboards*
<http://www.microsoft.com/downloads/details.aspx?familyid=1D3A4A0D-7E0C-4730-8204-E419218C1EFC&displaylang=en>
- *SQL Server Best Practice Analyzer*
<http://www.microsoft.com/downloads/details.aspx?FamilyId=DA0531E4-E94C-4991-82FA-F0E3FBD05E63&displaylang=en>
- *Microsoft SQL Server TechCenter*
<http://technet.microsoft.com/en-us/sqlserver/default.aspx>
- *Optimizing Server Performance*
<http://msdn2.microsoft.com/en-us/library/ms188284.aspx>
- *Microsoft SQL Server Performance Blog*
<http://blogs.msdn.com/sqlperf/>

Configuring the Disk Subsystem for Optimal Performance

With a physical disk that maintains 64 sectors per track, Windows always creates the partition starting at the sixty-fourth sector, therefore misaligning it with the underlying physical disk. To ensure proper disk alignment, use the diskpart.exe utility, provided with Microsoft Windows Server 2003 Service Pack 1, to explicitly set the starting offset in the master boot record (MBR).

Note: Diskpart.exe supersedes the functionality previously offered by Diskpar.exe.

Caution: Diskpart.exe is a data destructive utility that wipes all data from a disk during the storage track boundary alignment process. Therefore, if the disk on which you will run Diskpart contains data, backup the disk before performing the following procedure.

- ▶ **To align I/O with storage track boundaries using Diskpart.exe**
 1. If the disk you are aligning is blank (raw), proceed to Step 3.
 2. Backup any data If the disk contains data, backup the disk before proceeding.
 3. Delete all partitions on the disk.
 4. At a command prompt, type **diskpart.exe**
 5. At the Diskpart command prompt, type **List Disk** and press Enter. If the disk you want to align does not appear in the list, make sure that it exists and is accessible using the Disk Management snap-in.

6. Type **Select Disk X**, where *X* is the number of the disk as shown in the output of the **List Disk** command. Diskpart should return a message that indicates that Disk *X* is the selected disk.
7. Type **Create Partition Primary Align=X**, where *X* is either 32 or 64, depending on the recommendation from your storage vendor. If your storage vendor does not have any specific recommendations, it is recommended that you use 64.
8. Type **Assign Letter=<DriveLetter>**. For example, to assign letter Z to the disk, type **Assign Letter=Z**
9. After assigning the drive letter, exit the Diskpart utility, and then use the Disk Management snap-in or the Windows Format command to format the partition as an NTFS-formatted partition

Note: For more information about using the diskpart.exe utility, see the MSDN article *How to Align Exchange I/O with Storage Track Boundaries*:

[http://msdn2.microsoft.com/en-us/library/aa998219\(EXCHG.80\).aspx](http://msdn2.microsoft.com/en-us/library/aa998219(EXCHG.80).aspx)

Minimizing Locking Contention by Using Isolation Levels

Locking contention can adversely impact the performance of Microsoft SQL Server. In service scheduling scenarios, to minimize locking contention while protecting transactions from "dirty" reads of uncommitted data modifications, use either:

- READ COMMITTED isolation with the READ_COMMITTED_SNAPSHOT database option set to ON.
- SNAPSHOT isolation

Important: Using these isolation levels can also require higher levels of server processing to maintain the row versions in the temp database. Use server sizing as necessary to offset any impact on performance.

READ COMMITTED Isolation

READ COMMITTED isolation specifies that statements cannot read data that has been modified but not committed by other transactions. This prevents dirty reads. Data can be changed by other transactions between individual statements within the current transaction, resulting in non-repeatable reads or phantom data. This option is the SQL Server default.

With READ_COMMITTED_SNAPSHOT set to ON, the Database Engine uses row versioning to present each statement with a transactionally consistent snapshot of the data as it existed at the start of the statement. Locks are not used to protect the data from updates by other transactions.

SNAPSHOT Isolation

SNAPSHOT isolation specifies that data read by any statement in a transaction will be the transactionally consistent version of the data that existed at the start of the transaction. The transaction can only recognize data modifications that were committed before the start of the transaction. Data modifications made by other transactions after the start of the current transaction are not visible to statements executing in the current transaction. The effect is as if the statements in a transaction get a snapshot of the committed data as it existed at the start of the transaction.

Note: For more information about minimizing locking contention by using isolation levels, in SQL Server Books Online, view the following resources:

- *SET TRANSACTION ISOLATION LEVEL (Transact-SQL)*
<http://msdn2.microsoft.com/en-us/library/ms173763.aspx>
- *Adjusting Transaction Isolation Levels*
<http://msdn2.microsoft.com/en-us/library/ms189542.aspx>
- *Customizing Transaction Isolation Level*
<http://msdn2.microsoft.com/en-us/library/ms175909.aspx>

Disabling Support for Parallel Plan Queries

On computers with multiple processors, Microsoft SQL Server determines the optimal number of processors, degree of parallelism, required to run a single statement, for each parallel plan execution. Administrators can use the 'max degree of parallelism' option to limit the number of processors to use in parallel plan execution.

In a default configuration of Microsoft SQL server, the 'max degree of parallelism' value is set at 0, which specifies to use all available processors. Setting this value to 1 will suppress parallel plan generation, while setting the value to a number greater than 1 (up to a maximum of 64) will restrict the maximum number of processors used by a single query execution. If a value greater than the number of available processors is specified, the actual number of available processors is used. If the computer has only one processor, the 'max degree of parallelism' value is ignored.

This setting can be changed by using SQL Server Management Studio, or using the sp_configure system stored procedure.

Important: The 'max degree of parallelism' option is an advanced option. If you are using the sp_configure system stored procedure to change the setting, you can change 'max degree of parallelism' only when 'show advanced options' is set to 1. The setting takes effect immediately (without restarting the MSSQLSERVER service). For more information, in SQL Server Books Online, see the topic *max degree of parallelism Option*:
<http://msdn2.microsoft.com/en-us/library/ms181007.aspx>

For more information about setting the 'max degree of parallelism' value in SQL Server Management Studio, see *How to: Configure the Number of Processors Available for Parallel Queries (SQL Server Management Studio)*:
<http://msdn2.microsoft.com/en-us/library/ms189094.aspx>

Using Efficient Queries

When planning and implementing a query strategy, keep in mind the following guidelines for designing efficient queries.

- *Retrieve only the data required to meet a specific business requirement.*
Retrieving more data than actually required can lead to increased network traffic and require additional server and client resources.
- *Avoid mixing OLTP, OLAP, and reporting workloads.* OLTP workloads are characterized by many small transactions, with an expectation of very quick response time from the user, while OLAP and reporting workloads are characterized by a few long-running operations that might consume more resources and cause more contention. Avoid mixing these types of workloads if possible.

- *Avoid using specific types of transactions.* Avoid using long-running transactions, transactions that depend on user input to commit, transactions that never commit because of an error, and non-transactional queries inside transactions, which can lock resources that might otherwise be used to bolster overall performance.
- *Use efficient schemas.* Query inefficiencies can stem from poor schema design. This can result, for example, in too many or inefficient join operations. Schema design is largely a tradeoff between good read performance (improved through denormalization) and good write performance (improved through normalization).
- *Use indexes properly.* Create indexes to support the queries that are issued against your server, and avoid using too many indexes, which can affect the insert and update performance. Balance indexing needs according to business requirements.

Optimizing and Maintaining Query Performance

You can optimize query performance by partitioning a table or index, based upon the types of queries that are run frequently and the underlying hardware configuration.

Note: For more information about partitioning tables and indexes, and optimizing SQL query performance, in SQL Server Books Online, view the following resources:

- *Designing Partitions to Improve Query Performance*
<http://msdn2.microsoft.com/en-us/library/ms177411.aspx>
- *Query Tuning Recommendations*
<http://msdn2.microsoft.com/en-us/library/ms188722.aspx>

Partitioning for Join Queries

If you frequently run queries that involve an equi-join between two or more partitioned tables, their partitioning columns should be the same as the columns on which the tables are joined. Additionally, the tables, or their indexes, should be colocated. This means that they either use the same named partition function, or they use different ones that are essentially the same, in that they:

- Have the same number of parameters that are used for partitioning, and the corresponding parameters are the same data types
- Define the same number of partitions
- Define the same boundary values for partitions

In this way, the SQL Server query optimizer can process the join faster, because the partitions themselves can be joined. If a query joins two tables that are not colocated or are not partitioned on the join field, the presence of partitions may actually slow down query processing instead of accelerate it.

Taking Advantage of Multiple Disk Drives

It may be tempting to map your partitions to filegroups, each accessing a different physical disk drive, to improve I/O performance. When SQL Server performs data sorting for I/O operations, it sorts the data first by partition. Under this scenario, SQL Server accesses one drive at a time, and this might reduce performance. A better solution in terms of performance is to stripe the data files of your partitions across more than one disk by setting up a RAID. In this way, although SQL Server still sorts data by partition, it can access all the drives of each partition at the same time. This configuration can be designed regardless of whether all partitions are in one filegroup or multiple filegroups.

Maintaining Query Performance

Over time, the performance of existing queries may regress, or new queries may take longer than expected to complete. The degradation in performance can result from a number of causes, such as changes in statistical data that lead to a poor query plan for an existing query, missing indexes that force table scans, or an application slow down resulting from excessive blocking. Be sure to monitor query performance regularly to help maintain optimal performance.

Optimizing and Maintaining the Microsoft Dynamics CRM Database

The performance of the Microsoft Dynamics CRM database depends in part on the configuration of the physical design structures in the database, which include indexes, clustered indexes, indexed views, and partitions, whose purpose is to enhance performance and manageability of databases.

Note: For more information about optimizing and maintaining the Microsoft Dynamics CRM database and database indexes, in SQL Server Books Online, view the following resources:

- *Designing and Creating Databases*
<http://msdn2.microsoft.com/en-us/library/ms191425.aspx>
- *Optimizing Indexes*
<http://msdn2.microsoft.com/en-us/library/ms190910.aspx>

Segregating the Database and Transaction Log Files

Transaction log file creation can be write-intensive during periods when there is a high volume of data being added, changed, or removed from the application. For optimal performance, ensure that database files and transaction log files are located on separate sets of physical disks.

Optimizing and Maintaining Database Indexes

Maintaining indexes is a key factor in achieving minimum disk I/O for all database queries. The database maintenance command ALTER INDEX can help in de-fragmenting indexes in Microsoft SQL Server and in rebuilding one or more indexes for a specific table.

Tip: For large databases, consider creating indexes on separate file groups.

Note: For more information about ALTER INDEX, in SQL Server Books Online, view the following resources:

- *ALTER INDEX (Transact-SQL)*
<http://msdn2.microsoft.com/en-us/library/ms188388.aspx>
- *Performing Index Operations Online*
<http://msdn2.microsoft.com/en-us/library/ms177442.aspx>

Rebuilding Indexes

Rebuilding an index drops and re-creates the index. This removes fragmentation, reclaims disk space by compacting the pages based on the specified or existing fill factor setting, and reorders the index rows in contiguous pages. When ALL is specified, all indexes on the table are dropped and rebuilt in a single transaction. FOREIGN KEY constraints do not have to be dropped in advance. When indexes with 128 extents or

more are rebuilt, the Database Engine defers the actual page de-allocations, and their associated locks, until after the transaction commits.

Important: Repeatedly re-indexing the database to improve performance can ultimately lead to an unacceptable level of disk fragmentation. Consider the associated tradeoffs when determining how frequently to perform this task.

Reorganizing Indexes

Reorganizing an index uses minimal system resources. It defragments the leaf level of clustered and non-clustered indexes on tables and views by physically reordering the leaf-level pages to match the logical, left to right, order of the leaf nodes. Reorganizing also compacts the index pages. Compaction is based on the existing fill factor value.

Optimizing and Maintaining the CRM E-mail Router

Optimizing the CRM E-mail Router

To help ensure that the CRM E-mail router performs optimally within a specific Microsoft Dynamics CRM solution, keep the following points in mind:

- Implement the Microsoft Dynamics CRM E-mail router and the mailboxes it will process on the same server.
- Configure the Microsoft Dynamics CRM E-mail router to monitor forward mailboxes, and then modify the default forwarding rule to limit forwarded mail.
- For enterprise deployments, define multiple outgoing SMTP server profiles to provide users in each region with a unique SMTP server.

In addition, be sure to customize the default configuration of the E-mail router for optimal performance while maintaining overall business requirements. Consider the example scenarios provided in the following table.

Configuration Setting	Example Scenarios
Connection Timeout (seconds) (default = 300)	Because connections to some forward mailboxes can be unreliable, an administrator might configure the 'Connection Timeout (seconds)' setting to a low value to prevent the e-mail router from "hanging" on a particular mailbox.
Maximum Messages Per Cycle (default = 1000)	When configured to process multiple forward mailboxes, the e-mail router uses parallel processing to act on two mailboxes simultaneously. However, a single forward mailbox receiving a sudden blast of e-mails could potentially engage a thread for an unacceptable period of time. To introduce some level of fairness, an administrator might configure the 'Maximum Messages Per Cycle' setting to limit the number of e-mails the e-mail router would process in a single mailbox before automatically freeing up a thread to advance to the next forward mailbox.

Configuration Setting	Example Scenarios
Polling Period (seconds) (default = 60)	When configured to process multiple forward mailboxes, the e-mail router uses a parallel polling strategy. Each polling cycle consists of logging on, in sequence, to pairs of forward mailboxes, followed by a sleep. The 'Polling Period (seconds)' setting controls the duration of the sleep period. In general, increasing the duration of the sleep period decreases the need for system resources. However, if the e-mail router must process several forward mailboxes, a lengthy a sleep period may not be optimal.
Message Expiration (seconds) (default = 86400 [24 hours])	Administrators can configure the 'Message Expiration (seconds)' setting to control the period that the router will attempt to deliver an e-mail before moved it to the undeliverable folder. Note that if the value is set too low, a simple CRM server reboot may cause failures, while if the value is set too high, excessive retries may waste system resources.

Maintaining the CRM E-mail Router

To maintain the performance of the Microsoft Dynamics CRM E-mail router within an implementation, consider the following best practices.

- Frequently monitor the size of inbox folders to make sure the forward mailbox is not accumulating too many e-mails (possibly because of a spam attack, or because the e-mail router was shut down, etc.).
- For verbose logging, set the e-mail router service registry key 'LogLevel' (located under HKLM\System\CurrentControlSet\Services\MSCRMEmail) to a value of 4.
- Use Windows Performance Monitor (perfmon) and include the counters located under the performance object 'MSCRMEmail'.

Note: A complete list of the performance counters available for tracking e-mail processing and managing the overall performance of the CRM E-mail router in a Microsoft Dynamics CRM 4.0 implementation is provided in [Appendix B: CRM E-mail Router Performance Counters](#).

Appendix A: Additional Resources

Information Resources

For more information about optimizing and maintaining Microsoft Dynamics CRM 4.0, view the following resources:

- Microsoft Dynamics CRM 4.0 Software Development Kit
<http://www.microsoft.com/downloads/details.aspx?FamilyId=82E632A7-FAF9-41E0-8EC1-A2662AAE9DFB&displaylang=en>
- Microsoft Dynamics CRM 4.0 Implementation Guide
<http://www.microsoft.com/downloads/details.aspx?FamilyId=1CEB5E01-DE9F-48C0-8CE2-51633EBF4714&displaylang=en>
- Microsoft Dynamics CRM Developers Center
<http://msdn2.microsoft.com/en-us/dynamics/crm/default.aspx>
- Microsoft Dynamics CRM 4.0 Server Readme
<http://www.microsoft.com/downloads/details.aspx?FamilyId=1DB2D16D-7885-4561-A461-20B948913B3F&displaylang=en>
- Microsoft Dynamics CRM 4.0 for Microsoft Office Outlook Readme (On-Premise and Hosted Editions)
<http://www.microsoft.com/downloads/details.aspx?FamilyId=18D51D2E-F82B-46CF-8CAC-DF06A5F6555F&displaylang=en>

Note: The Microsoft Dynamics CRM 4.0 Performance and Stress Testing Toolkit will become available subsequent to the release of this document. In the interim, please use the Microsoft Dynamics CRM 3.0 Performance and Stress Testing Toolkit available at: <http://www.microsoft.com/downloads/details.aspx?familyid=1a25db7c-5060-417c-86db-6377a84ee650&displaylang=en>

Important: For the latest information about Microsoft Dynamics CRM 4.0 performance and to learn about the release of additional support resources, visit the Microsoft Dynamics CRM Team blog (<http://blogs.msdn.com/crm/>).

Technical Services

Consider the following resources for technical services beyond the scope of those offered by the Microsoft Dynamics CRM Technical Presales Advisory Group (TPAG) as outlined at the following PartnerSource Web site: <https://partner.microsoft.com/global/40023009>

Note: For services outside of North America and the United Kingdom, contact your local Microsoft Support Services or Microsoft Consulting Services (MCS) office to request hardware sizing or architecture services (availability may vary).

North America – Business Systems Architecture Services

The North America Microsoft Dynamics Business Systems Architecture Team provides prescriptive guidance on deployment infrastructure and hardware to partners and customers for Microsoft Dynamics deployments. Specific packaged services include Business Systems Architecture Assessments (includes hardware sizing), Health Checks, and Onsite System Performance Workshops. For pricing information and availability, contact MBSProfessionalServices@microsoft.com.

To learn more about the services provided by the Microsoft Dynamics Business Systems Architecture Team, see <https://partner.microsoft.com/US/40029785>.

United Kingdom - UK Microsoft Dynamics Consulting

The UK Microsoft Dynamics Consulting team offers a full range of services to address the complete project life cycle, including help with complex sizing, performance tuning, or load testing scenarios, a series of workshops, assistance, and quality assurance offerings are available. For pricing information and availability, contact ukcrmc@microsoft.com.

Appendix B: CRM E-mail Router Performance Counters

The following table lists the performance counters available for tracking e-mail processing and managing the overall performance of the CRM E-mail router in a Microsoft Dynamics CRM 4.0 implementation.

Performance Counter	Tracks the...
Incoming e-mail messages delivered	Total number of incoming e-mail messages delivered successfully.
Incoming e-mail messages discarded	Total number of incoming e-mail messages not accepted for delivery.
Incoming e-mail messages potentially corrupted	Total number of incoming e-mail messages that are potentially corrupted.
Incoming e-mail messages processed	Total number of incoming e-mail messages processed.
Incoming e-mail messages processed per second	Number of incoming e-mail messages processed per second.
Incoming e-mail messages undelivered	Total number of incoming e-mail messages not delivered successfully.
Incoming mailbox access attempt failures	Total number of unsuccessful mailbox access attempts for incoming e-mail messages.
Incoming mailbox access attempts	Total number of successful mailbox access attempts for incoming e-mail messages.
Outgoing e-mail messages delivered	Total number of outgoing e-mail messages delivered successfully.
Outgoing e-mail messages processed	Total number of outgoing e-mail messages processed.
Outgoing e-mail messages processed per second	Number of outgoing messages processed per second.
Outgoing e-mail messages undelivered	Total number of outgoing e-mail messages not delivered successfully.
Service configuration refreshes	Total number of times the configuration was refreshed.
Service configuration scheduling cycles	Total number of times a scheduling cycle occurred.
Service provider load failures	Total number of times a service provider could not be loaded.
Service providers aborted	Total number of times a service provider was aborted because it took too long to execute.
Service providers executed	Total number of times a service provider completed its work.
Service providers failed	Total number of times a service provider failed during execution.
Service providers refreshed	Total number of times a service provider was changed and had its configuration refreshed.
Service providers removed	Total number of times a service provider was deleted from the configuration and removed from the schedule.
Service providers started	Total number of times a scheduled service provider was started.