
# Optimizing and maintaining the performance of a Microsoft Dynamics CRM 2011 server infrastructure

Microsoft Corporation

Published: August 2011

Updated: September 2013

## Abstract

Performance tuning and optimization is a continuous balancing act between design decisions and resource availability. This white paper provides tips, tricks, and guidance for optimizing and maintaining the performance of a Microsoft Dynamics CRM 2011 server infrastructure.

Microsoft

# Contents

# Optimizing and maintaining a Microsoft Dynamics CRM 2011 server infrastructure

**Contributors**: Brian Bakke, Grant Geiszler, Martijn Bronkhorst

**Technical Reviewers**: Gus Apostol, Ramani Jagadeba, Mahesh Vijayaraghavan, Chandra Akkiraju

Published: August 2011  Updated: September 2013

Performance tuning and optimization is a continuous balancing act between design decisions and resource availability. This white paper provides tips, tricks, and guidance for optimizing and maintaining the performance of a Microsoft Dynamics CRM 2011 server infrastructure.

## Applies To

- Microsoft Dynamics CRM 2011

**In this White Paper**



This section introduces the purpose, scope, and applicability of the information provided in this paper.

**Purpose**

This white paper complements, rather than replaces, existing resources that are specific to optimizing and maintaining the various components that make up a Microsoft Dynamics CRM 2011 implementation. To that end, this document provides relevant context for each component, information on maintaining and optimizing that component specifically within the Microsoft Dynamics CRM implementation, and links to related resources that may offer additional guidance for particular environments.

**Scope**

This paper focuses on optimization and maintenance of the server infrastructure associated with a business solution based on an on-premises or hosted deployment of Microsoft Dynamics CRM 2011. For information focused specifically on optimizing and maintaining the Microsoft Dynamics CRM clients that connect to an instance of Microsoft Dynamics CRM 2011 or Microsoft Dynamics CRM Online, see the white paper Optimizing and maintaining client performance for Microsoft Dynamics CRM 2011 and CRM Online.

**Applicability**

When considering the applicability of the information in this white paper to any specific solution based on Microsoft Dynamics CRM 2011, note that the techniques and guidance provided in this white paper can yield varying results depending on a wide range of potential environmental factors, for example the level and complexity of the customizations applied to a Microsoft Dynamics CRM solution.

As a result, be sure to verify the functionality and performance impact associated with any specific optimization technique in a test environment prior to making any change in a production environment.

**Important**
For Microsoft Dynamics CRM 2011 deployments that are integrated with other systems, be sure to verify optimization techniques in a test environment that approximates the complexity and integration that is present in the production environment.

**Download**

This paper can be downloaded from the Microsoft Download Center: Optimizing and maintaining a Microsoft Dynamics CRM 2011 server infrastructure.

# Overview of optimizing a server infrastructure

Microsoft Dynamics CRM 2011 requires several software applications and components that work together to create an effective system. In addition, the product supports a variety of server topologies.

For example, you can deploy Microsoft Dynamics CRM 2011 solution in a:

- Single-server configuration, with Microsoft Dynamics CRM Server 2011, SQL Server, SQL Server Reporting Services, and optionally, Microsoft Exchange Server installed and running on the same computer.
- Multiple-server configuration, with a dedicated server for each of:
  - Microsoft Dynamics CRM Server 2011
  - Windows Server (Active Directory domain controller/Internal DNS)
  - Microsoft SQL Server (Database server)
  - Microsoft Dynamics CRM 2011 Email Router or Microsoft Exchange Server

The deployment architecture that is appropriate for any specific implementation of Microsoft Dynamics CRM 2011 depends on a variety of criteria, including the specific business needs that the solution is designed to address.

Successfully optimizing and maintaining the performance of the server infrastructure that supports a business solution based on Microsoft Dynamics CRM 2011 requires:

- Familiarity with:
  - The general approach to optimization and maintenance
  - The server architecture and components that make up a Microsoft Dynamics CRM implementation
  - Factors that can adversely affect the performance of Microsoft Dynamics CRM system components or the interaction among those components
- A solid understanding of:
  - Options available for configuring the components of a Microsoft Dynamics CRM 2011 solution to overcome potential performance issues
  - Best practices for customizing a Microsoft Dynamics CRM 2011 solution for optimal performance

This white paper addresses these topics, providing information that helps readers achieve and maintain optimal performance of the server infrastructure supporting a Microsoft Dynamics CRM 2011-based business solution deployed in an on-premises or hosted environment.

# General approach to optimization and maintenance

Performance tuning is a continuous balancing act among design decisions, resource availability, cost, and power considerations. Consider taking the following high-level approach when devoting efforts to optimizing and maintaining a connected system such as Microsoft Dynamics CRM 2011.

**Important**

Regularly downloading and installing the latest hot fixes and update roll-ups is a critical first step to enhancing and maintaining the performance of an overall deployment. For a listing of current resources, updates, and hot fixes available for Microsoft Dynamics CRM 2011, on the Microsoft Support site, see the Support for Dynamics CRM page.

Also be sure to regularly download and install the latest updates for each of the system components (such as Windows Server and Microsoft SQL Server) that make up the solution.

## Optimizing performance and establishing a baseline

After initially deploying Microsoft Dynamics CRM 2011, it is important to optimize the overall 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.

## Monitoring and maintaining performance

Over the course of time, routine, daily use of a connected system such as Microsoft Dynamics CRM 2011 can adversely impact solution performance. After deploying a solution based on Microsoft Dynamics CRM 2011, be sure to leverage the functionality of the Windows Reliability and Performance Monitor to collect and analyze performance data that you can use to monitor the overall health of the system.

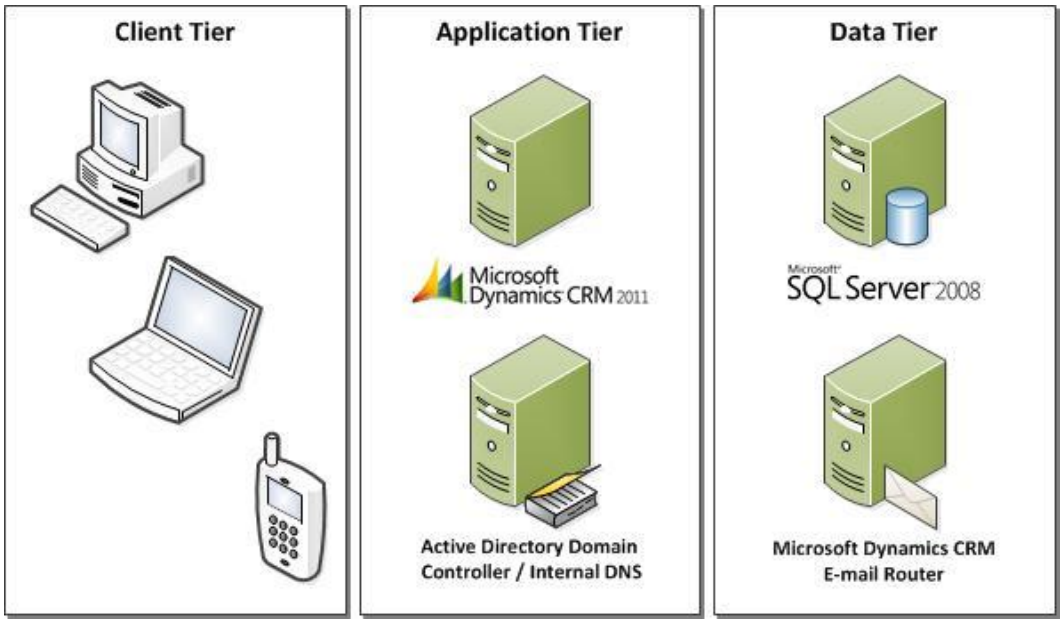
**Note**

- For more information about the Windows Reliability and Performance Monitor and how to use it effectively, see the TechNet article Performance and Reliability Monitoring Step-by-Step Guide for Windows Server 2008.
- For a complete listing of the performance counters available with Microsoft Dynamics CRM 2011, on the Microsoft Download Center, see the white paper Microsoft Dynamics CRM 2011 Performance Counters.
- For additional information about general guidelines and best practices for monitoring performance, on TechNet, see the article Performance Monitoring Getting Started Guide.
- For the latest information about Microsoft Dynamics CRM performance in general and to learn about the release of additional support resources, visit the Microsoft Dynamics CRM Team blog.

# Logical architecture for Microsoft Dynamics CRM 2011

Understanding the unique aspects and system components that are generally associated with the logical architecture for implementing a Microsoft Dynamics CRM 2011 solution is critical to being successful at optimizing and maintaining the performance of the server infrastructure that supports the overall system.

While there are a variety of specific hardware and software requirements for implementing a solution based on Microsoft Dynamics CRM 2011, deployments in general follow a similar, logical architecture that includes three tiers, as shown in the following graphic.




**Important**

The graphic shows one example of the architecture for implementing Microsoft Dynamics CRM 2011; this design is not intended to be prescriptive in any way. Planning an implementation to address a specific business environment may indicate that one of many alternate structures would be more appropriate in a particular situation.

## The client tier

The client tier provides for interactions between the Microsoft Dynamics CRM 2011 system and its users, and tier components are configured and optimized for responsive user interaction. The system components that typically fall within the client tier include the following:

- Microsoft Dynamics CRM for Outlook
- Web Client
- Mobile Express Client

Potential performance issues affecting the client tier include network link issues (ping time, cache size and expiration time, compression technology) or relate to offline synchronization.

**Important**

For techniques related to optimizing and maintaining the performance of the clients connecting to a Microsoft Dynamics CRM deployment, see the white paper Optimizing and Maintaining Client Performance for Microsoft Dynamics CRM 2011 and CRM Online.

## The application tier

The application tier centralizes and performs all business logic processing, as well as providing the Web server responsibilities, for a Microsoft Dynamics CRM 2011 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 more difficult than optimizing the client tier.

The system components that typically fall within the application tier include the following:

- Windows Server 2008 R2
- Microsoft .NET Framework and Microsoft .NET Applications
- Internet Information Services
- Microsoft Dynamics CRM Server 2011
  - Front End Server
  - Back End Server (CRM Async Service)
  - Deployment Administration Server
- Microsoft Dynamics CRM Reporting Services

Potential performance issues that commonly affect the application tier include resource bottlenecks (processor, memory) or relate to plug-ins, SDK code, or workflow.

## 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 2011 implementation includes both Microsoft SQL Server and the databases containing the records and information unique to an organization. Depending on the implementation, the data tier can also include the Microsoft Dynamics CRM Email Router, as well as the Email server and database.

The system components that typically fall within the data tier include:

- Microsoft SQL Server
- Microsoft Dynamics CRM Database
- Microsoft Dynamics CRM Email Router or Microsoft Exchange Server

Potential performance issues that commonly affect the data tier relate to physical data layout, the storage subsystem, or indexes.

# Server-side techniques for optimizing the client tier

While there are a variety of techniques available for configuring and customizing the clients connecting to a deployment of Microsoft Dynamics CRM 2011 or Microsoft Dynamics CRM Online for optimal performance, there are also server-side settings that can help to enhance the performance of Microsoft Dynamics CRM clients.

# Using compression techniques

Compression techniques designed to optimize network performance directly affect the size of the data files transmitted over the network.

## Configuring HTTP compression

By using bandwidth more effectively, enabling compression can provide faster transmission times between IIS and compression-enabled browsers. If your network bandwidth is restricted, as it is, for example, with mobile phones, compression can improve performance. IIS can be configured to compress only static files, only dynamic application responses, or both static files and dynamic application responses.

**Important**

HTTP compression is usually available in a default installation of IIS 7. However, only static compression is installed by default. For information about how to install static or dynamic compression, see **Setup** in the article HTTP Compression <httpCompression>.

**To configure HTTP compression**

1. On the Microsoft Dynamics CRM Server, in the Internet Information Services (IIS) Manager administrative tool, expand **Sites**, and then click **Microsoft Dynamics CRM**.
2. In **Features View**, double-click **Compression**, and then specify whether to compress dynamic content, static content, or both.
3. If you enable compression of static content, then under **Static Compression**:
   a. In the **Only compress files larger than (in bytes)** text box, type the minimum file size that you want IIS to compress; the default size is 256 bytes.
   b. In the **Cache directory** text box, type the path of a local directory or click the browse button **. . .** (three dots) to locate a directory. After a static file is compressed, it is cached in this temporary directory until it expires, or until the content changes. The temporary directory must be on a local drive on an NTFS-formatted partition. The directory cannot be compressed and should not be shared.
   c. Optionally, select the box next to **Per application pool disk space limit (in MB)** and type the maximum amount of space per application pool, in megabytes, you want IIS to use when it compresses static content. For example, if there are 20 application pools on the server and the **Disk space limit** is set to 100, the maximum disk space will be 2GB. If you click the **Per application pool disk space limit (in MB)** option and type a number in the text box under it, IIS automatically cleans up the temporary directory according to a least recently used rule when the set limit is reached; the default is 100 MB per application pool.
4. In the **Actions** pane, click **Apply**.

**Note**

For additional information about configuring HTTP compression, on TechNet, see the article Configuring HTTP Compression in IIS.

By default, Microsoft Dynamics CRM 2011 is configured to compress web responses that are sent to browser clients. However, Microsoft Dynamics CRM 2011 is not by default configured to compress HTTP responses sent to SDK clients because this is an IIS-wide setting that can't be configured at the Microsoft Dynamics CRM site level.

Regardless, enabling HTTP compression can have a great impact on the performance of Microsoft Dynamics CRM for Outlook, which is an SDK client. An example of the impact compression can have on network resource utilization for a vanilla, out of box CRM server with sample data is shown in the following table, which provides a comparison of the bytes transferred when switching to Accounts View:

| | **No Compression** | **Compression Enabled on HTTP** | **Compression Enabled on HTTPS** |
|---|---|---|---|
| Bytes Sent | 105084 | 105084 (0% reduction) | 67586 (36% reduction) |
| Bytes Received | 219102 | 149424 (32% reduction) | 25837 (88% reduction) |

Note that using HTTPS with compression shows an even more significant improvement in network resource utilization over than does using HTTP with compression.

To configure Microsoft Dynamics CRM Server 2011 to compress the HTTP responses that are sent to SDK clients, turn on IIS dynamic compression for the SOAP responses that are returned by Microsoft Dynamics CRM server by creating a batch file named **IIS7-EnableSOAPXML-DynamicsCompression.cmd** and adding to it the following line (one line):

```
%SYSTEMROOT%\system32\inetsrv\appcmd.exe set config -
section:system.webServer/httpCompression
/+"dynamicTypes.[mimeType='application/soap%%u002bxml; charset=utf-8',enabled='true']"
/commit:apphost
```

Finally, run the batch file from an administrative command prompt on the Microsoft Dynamics CRM server, and then perform an IIS Reset at a maintenance window to ensure that the setting is applied.

**Important**

To turn off this compression, simply create another batch file, IIS7-RemoveSOAPXML-DynamicsCompression.cmd and add to it the following line (one line):

```
%SYSTEMROOT%\system32\inetsrv\appcmd.exe set config -
section:system.webServer/httpCompression/-
"dynamicTypes.[mimeType='application/soap%%u002bxml; charset=utf-
8',enabled='true']" /commit:apphost
```

Note that if you are running Windows Server 2008 and with the IIS 7 Administration Tools pack (the Tools pack is installed by default and built into IIS 7.5 with Windows Server 2008R2), you

can also manually configure IIS dynamic compression (or view the configuration to confirm the setting value).

**To manually configure IIS dynamic compression**

1. In Internet Information Services Manager, click the <Server name>, scroll down in the Features View to the **Management** section, and then start the Configuration Editor.
2. In the Configuration Editor, in the **Section: address** box, type **system.webServer/httpCompression**, click **dynamicTypes**, and then in the value cell, click the ellipses **. . .** (three dots).
3. In the Collection Editor, click **Add** and verify that **Enabled** is set to **True** and the inclusion of an entry for **mimeType of application/soap+xml; charset=utf-8**.Note that the appearance here is different than it was when using the **appcmd** command, which has URL encoded the plus symbol.
4. Click one of the **Items: collection** cells, and then close the Collection Editor.
5. In the Configuration Editor, under **Actions**, click **Apply** to add configure the new setting **httpCompression mimeType**.
6. Run an IISReset to ensure that the compression setting is enabled for the new mimeType.

**Warning**
For additional information, see the article HTTP Compression <httpCompression>.

### Using WAN accelerator hardware

To address latency issues, several vendors offer web-acceleration appliances that improve the performance of applications such as Microsoft Dynamics CRM, and using these devices can greatly improve Microsoft Dynamics CRM performance over the WAN. Several Microsoft Dynamics CRM customers have implemented WAN acceleration appliances with encouraging results. Using WAN accelerator hardware can help improve performance especially in on-premises scenarios with a geographically distributed deployment in which users are distributed around the world and performance for users in a specific location is not satisfactory.

## Configuring proxy server settings

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 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-premises deployments of Microsoft Dynamics CRM 2011 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, see the following resources:

- Using Automatic Proxy Configuration.
- Automatic Proxy Detection.
- How to disable automatic proxy caching in Internet Explorer.
- How to configure client proxy server settings by using a registry file.
- You experience a delay when you use your Windows XP computer to log on to a domain or to connect to a network resource.
- Change proxy settings in Internet Explorer.

## Reducing email traffic by modifying Outlook rules

For deployments with Microsoft Exchange Server or the Microsoft Dynamics CRM Email 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 only track Microsoft Dynamics CRM emails, configure Outlook to forward only messages that include **CRM:** in the subject or body.

**Warning**

**CRM:** is the default prefix specified for the tracking token in the System Settings > Email tracking tab in Microsoft Dynamics CRM. 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 overfill should the Exchange Router service quit functioning.

**To modify the Outlook email forwarding rule**

1. In Outlook 2003 or Outlook 2007, on the **Tools** menu, click **Rules and Alerts**.
   - OR -
   In Outlook 2010, on the **File** tab, on the **Info** panel, click **Manage Rules & Alerts**.
2. In the **Rules and Alerts** dialog box, on the **Email Rules** tab, select the existing Microsoft Dynamics 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)**, and 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**

For additional information about optimizing and maintaining the Microsoft Dynamics CRM Email Router, see the Optimizing the Data Tier section of this white paper. For information about optimizing and maintaining Microsoft Exchange Server, in the Exchange Server TechCenter, see the following resources:

- Microsoft Exchange Server 2010: Performance and Scalability.
- Exchange Server 2007: Monitoring Exchange 2007 with Microsoft Operations Manager 2005 SP1.
- Exchange Server 2003: Performance and Scalability Guide for Exchange Server 2003.

# Optimizing and maintaining the application tier

There are a number of techniques available for optimizing and maintaining the importance of the application tier, which are explained in the following sections.

## Optimizing and maintaining Windows Server

Proper configuration of the Windows Server operating system is a critical step in ensuring the optimal performance of any Microsoft Dynamics CRM 2011 deployment.

### Optimizing the performance of Windows Server

Before optimizing the performance of Windows Server 2008 within a broader Microsoft Dynamics CRM 2011 server infrastructure, it is important to follow and apply component specific guidelines and techniques to ensure that the operating system itself is performing at optimal levels

### Configuring Windows Server for optimal performance

Windows Server includes a variety of settings that can be configured to incrementally improve performance. These settings can be categorized according to area targeted for performance improvement. For example, specific settings are available in Windows Server 2008 for optimizations related to networking, storage, IIS, file servers, or Active Directory.

**Warning**

For more information about configuring Windows Server 2008 settings for optimal performance, see the following resources:

- Performance Tuning Guidelines for Windows Server 2008 R2.
- Windows Server 2008 TechNet Center.

## Considerations for 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. In rare cases there may be no free ephemeral ports available, which will cause the connection open to fail or time out. While this would be unlikely to occur on a computer running in the client tier, it is more likely to affect a computer in the application tier that is creating connections for every client request that is processed.

With Windows Server 2008 (and Windows Vista), Microsoft has increased the dynamic client port range for outgoing connections; the new default start port is 49152 and the default end port is 65535, which replaces the previous default port range of 1025 through 5000 that was used for older versions of Windows operating systems.

**Important**
After installing Microsoft Exchange Server 2007 on a computer running Windows Server 2008, the default port range is 1025 through 60000.

**Note**
For more information about this change in the dynamic client port range for outgoing connections, on Microsoft TechNet, see the article The default dynamic port range for TCP/IP has changed in Windows Vista and in Windows Server 2008.

Also remember that the range is set separately for each transport and each version of IP. Microsoft customers who deploy servers running Windows Server 2008 may have problems with RPC communication between servers if firewalls are used on the internal network.

In these cases, it is recommended to reconfigure the firewalls to allow for traffic between servers in the dynamic port range of 49152 through 65535. This range is in addition to well-known ports that are used by services and by applications. Or, the port range that is used by the servers can be modified on each server.

**To view the current range of ephemeral ports**

- At a command prompt, type the following command, and then press Enter:

```
netsh int <ipv4|ipv6> show dynamicport <tcp|udp>
```

**To change the current range of ephemeral ports**

- At a command prompt, type the following command, and then press Enter:

```
netsh int <ipv4|ipv6> set dynamic <tcp|udp> start=number num=range
```

  In this command, *number* represents the number of the starting port, while *range* represents the total number of ports in the range.

**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 by using the following information:

| | |
|---|---|
| Value Name | MaxUserPort |
| Value Type | DWORD |
| Value Data | 65534 |
| Value Range | 5000-65534 (decimal) |
| Default | 0x1388 (5000 decimal) |
| Description | Controls the maximum port number 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.<br>**Warning**<br>After release of security bulletin MS08-037, the behavior of Windows Server 2003 was changed to more closely match that of Windows Server 2008 and Windows Vista. For more information about security bulletin MS08-037, click the following article numbers to view the associated Knowledge Base articles:<br>1. 951746 - MS08-037: Description of the security update for DNS in Windows Server 2008, in Windows Server 2003, and in Windows 2000 Server (DNS server-side): July 8, 2008.<br>2. 953230 - MS08-037: Vulnerabilities in DNS could allow spoofing. |

3. Close the Registry Editor, and then restart the computer.

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

**Warning**

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

## Monitoring the performance of Windows Server

You can use 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.

**Warning**

For more information about monitoring the performance of Windows Server 2008, see the TechNet article Performance and Reliability Monitoring Step-by-Step Guide for Windows Server 2008.

## Optimizing and maintaining the Microsoft .NET Framework and Microsoft .NET applications

The following sections provide details about how to tune the performance of 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.

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. |

| Potential Issue | Description |
| --- | --- |
| 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. |
| 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, see the following resources:

1. Chapter 5 — Improving Managed Code Performance.
2. Checklist: Managed Code Performance.

## 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 | This can result from poorly managed or unmanaged memory. To identify this symptom, watch the following performance counters:<br>• Process\Private Bytes<br>• .NET CLR Memory\# Bytes in all Heaps<br>• Process\Working Set<br>• .NET CLR Memory\Large Object Heap size<br>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 | This refers to the set of memory pages that currently loaded in RAM, which is measured by “Process\Working Set”. A high value might indicate that several assemblies are loaded. Unlike other counters, “Process\Working Set” has no specific threshold value to watch, though a high or fluctuating value can indicate a memory shortage, especially if accompanied by a high rate of page faults. |
| 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 a result of poorly written managed code that:<br>• Causes excessive garbage collection (measured by “% Time in GC”)<br>• Throws many exceptions (measured by “.NET CLR Exceptions\# of Excepts Thrown /sec.”) |

| Bottleneck | Description |
| --- | --- |
| | • 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:<br>• .NET CLR LocksAndThreads\Contention Rate / sec<br>• .NET CLR LocksAndThreads\Total # of Contentions<br>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, see the following resources:

- Performance Counters for ASP.NET.
- Monitoring ASP.NET Application Performance.

# Optimizing and maintaining Internet Information Services

The following sections provide detailed about techniques for 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 2011 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 2011 SDK.

**Note**
For more information about optimizing IIS performance, see the following resources:

- IIS 7.0 Configuration Reference.
- IIS 7.0 Output Caching.
- IIS 7.0 Performance Discussions.

## Optimizing the performance of Integrated Windows Authentication and Kerberos authentication

Ensuring Kerberos is enabled at the client and the server can enhance performance by reducing the number of round trips required for authentication. When using Kerberos, the client can send authentication details with the initial response rather than having to go through multiple challenges and responses, which would otherwise be required. As a result, maintaining credentials across sessions can provide better performance, particularly on higher latency networks.

While Kerberos generally should “just work,” here are some scenarios or issues you may encounter.

- In Internet Explorer, ensure that “Enable Integrated Windows Authentication” is set to enable use of Kerberos for integrated authentication. The client will not attempt Kerberos authentication unless this setting is enabled.
- There is a known issue where users are members of too many groups to be communicated in a UDP packet. It is possible to get Kerberos to use TCP rather than UDP, which enables larger packets of information.
- To determine whether users are connecting via Kerberos, in the registry, enable Kerberos logging. If necessary, to help diagnose whether the user can obtain and use a Kerberos ticket, use applications such as Kerbtray, which is available from Microsoft Downloads.

While there are a variety of possible causes of Kerberos failure, enabling this feature can provide performance benefits, especially in low bandwidth environments. However, it is also important to take into account the need to perform network diagnostics to verify the ultimate impact on performance.

**Warning**

For more information about optimizing the performance of Windows Integrated authentication and Kerberos authentication, see the Knowledge Base article You may experience slow performance when you use Integrated Windows authentication together with the Kerberos authentication protocol in IIS 7.0.

## Additional optimization considerations

When optimizing the performance of IIS on the computer running Microsoft Dynamics CRM 2011 Server, keep in mind that 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 n 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.

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.

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.

## 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, collect information about several Windows Server 2008 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 IIS performance, see the following resources:

- New worker process performance counters in IIS7.
- Microsoft IIS Web site.

## Optimizing the performance of Microsoft Dynamics CRM Server 2011

To configure Microsoft Dynamics CRM 2011 for optimal performance, focus attention on both the Microsoft Dynamics CRM Server 2011 and on any Microsoft Dynamics CRM 2011-based applications running as part of the business solution.

**Important**

Disabled by default, Microsoft Dynamics CRM Server 2011 platform tracing can be useful in troubleshooting efforts, but enabling the feature in a production environment can negatively affect the performance of the server running Microsoft Dynamics CRM 2011. To verify that this feature is disabled, in the Registry Editor, navigate to the following location and check that the value of the “TraceEnabled” registry key is set to 0 (zero): HKEY_LOCAL_MACHINE | Software | Microsoft | MSCRM

**Note**

To turn off tracing at the deployment-level, you can use the PowerShell applet Set-CrmSetting. For additional information, on the Microsoft Support site, see the Knowledge Base article How to enable tracing in Microsoft Dynamics CRM.

### Hardware and software requirements

Before you consider optimizing the performance of servers running Microsoft Dynamics CRM Server 2011, verify that the computers meet the hardware and software requirements.

**Note**

For a listing of the hardware and software requirements for running Microsoft Dynamics CRM Server 2011, in the Implementation Guide, in the Planning Guide, see the following topics:

- Microsoft Dynamics CRM Server 2011 hardware requirements.
- Microsoft Dynamics CRM Server 2011 software requirements.

## Enhancing performance by distributing server roles on multiple servers

To take advantage of additional performance and scaling benefits in enterprise deployments, consider distributing specific server functionality, components, and services, or server roles, on different servers. Microsoft Dynamics CRM 2011 supports distribution and scaling of server roles across multiple servers. Having multiple servers also allows for implementation of departmental Microsoft Dynamics CRM systems that use the same Microsoft Dynamics CRM database.

Microsoft Dynamics CRM 2011 server roles are grouped into three categories:

- Front End Server
- Back End Server
- Deployment Administration Server

### Front End Server roles

The Front End Server role group includes the server roles for running client applications and applications developed with the Microsoft Dynamics CRM Software Development Kit (SDK).

Front End Server Roles in Microsoft Dynamics CRM 2011 are described in the following table.

| Server Role | Description |
|---|---|
| Web Application Server | Allows users to access data and content through the Web application client and Microsoft Dynamics CRM for Outlook. When you add this role, the Organization Web Service will also be added. |
| Organization Web Service | Provides the components necessary to run applications that use the methods described in the Microsoft Dynamics CRM SDK. |
| Discovery Web Service | Provides the components that let users find the organization to which they belong. |
| Help Server | Provides the components necessary to make Microsoft Dynamics CRM Help available to users. |

### Back End Server roles

The Back End Server role group includes the server roles that handle processing asynchronous events such as workflows and custom plug-ins. These roles are usually not exposed to the Internet. Back End Server Roles in Microsoft Dynamics CRM 2011 are described in the following table.

| Server Role | Description |
|---|---|
| Asynchronous Processing Service | Provides the components that are used to process queued asynchronous events such as bulk email or data import. |
| Sandbox Processing | Provides an isolated environment to allow for the execution of custom code, such as plug-ins, which reduces the possibility of custom code affecting the operation of the organizations in the Microsoft Dynamics CRM deployment. |

### Deployment Administration Server roles

The Deployment Administration Server role group provides the server roles for components that are used to manage the Microsoft Dynamics CRM deployment either by using the methods described in the Microsoft Dynamics CRM Deployment Software Development Kit (Deployment SDK) or the Deployment Manager. Deployment Administrator Server Roles in Microsoft Dynamics CRM 2011 are described in the following table.

| Server Role | Description |
|---|---|
| Deployment Tools | Includes Deployment Manager, which is a Microsoft Management Console (MMC) tool that deployment administrators can use to manage Microsoft Dynamics CRM organizations, servers, and licenses. |
| Deployment Web Service | Provides the components that are required to manage the deployment by using the methods described in the Microsoft Dynamics CRM Deployment SDK, such as for creating an organization or for removing a Deployment Administrator role from a user. |

### Considerations for distributing server roles

It is recommended to install server roles by group. However, a single server role or combination of roles can be installed on any Microsoft Dynamics CRM Server in the deployment. For example, a customer with a large user base might install the Front End Server role group on each of two or more servers running IIS to increase throughput performance for users.

**Important**

Server role groups can be installed on a server independent of any other server role group. However, all server roles must be running in your organization's network to provide a fully functioning system. If one or more server roles are missing in a deployment, Deployment Manager displays a message in the Messages area.Often, the best solution is to deploy all roles to all servers and then to simply enable the service for the role, such as the Async service, on the servers on which you want to run this and disable the service on the servers that are being used for load balancing.  This technique allows for scaling out a specific server role without having to reinstall anything.

**Note**

For more information about implementing Microsoft Dynamics CRM 2011 on multiple servers, in the Microsoft Dynamics CRM 2011 Implementation Guide, see the following topics:

- Install Microsoft Dynamics CRM Server 2011 on multiple computers.
- Multiple-server deployment.
- Advanced deployment options for Microsoft Dynamics CRM Server 2011.

## Throttling client synchronization processes

Microsoft Dynamics CRM 2011 provides deployment settings that allow you to throttle the client synchronization processes to improve overall performance.

**Note**

For additional information about the throttle settings available with Microsoft Dynamics CRM 2011, on MSDN, see the topic ThrottleSettings Class.

To change throttle settings, use the CRM PowerShell cmdlet **Set-CrmSetting**.

**Note**

For additional information about using the Set-CrmSetting cmdlet to change throttle settings, on MSDN, see the topic Use PowerShell to Call the Deployment Web Service.

## Limiting the number of records returned by aggregate queries

You can control the maximum number of records that are returned by aggregate queries (used in charts) from the Application tier by changing the advanced configuration setting **AggregateQueryRecordLimit**. This setting is stored in the DeploymentProperties table in the MSCRM_CONFIG database

For more information about limiting the number of records returned by aggregate queries, on TechNet, see the following resources:

- Deployment Table Metadata (Advanced Settings).
- Use Advanced Configuration Settings (ConfigDB).

## Optimizing the performance of queries against large datasets

For Microsoft Dynamics CRM 2011 business solutions that include an entity with a large dataset, record retrieval and grid rendering performance can be negatively impacted for users that do not have global Read access to all records associated with that entity. If there are significant differences in the levels of performance experienced by users with global access and users with more restricted access, consider modifying one or more system settings to ensure an optimal experience for all users.

### Best practices

To help ensure record retrieval and grid rendering performance is optimal for users that do not have global Read access to all of the data being queried, consider the following best practices:

- The more records that are shared with a user, the greater the potential that the user will experience a performance impact. Be judicious with the amount of sharing that occurs within a specific implementation – share as few records as possible while still accomplishing overall business goals.
- The more teams a user belongs to, the greater the potential that the user will experience a performance impact. Balance the need to limit the number of teams that a user belongs to with the need to develop a manageable business unit structure.

### Optimization techniques

With Microsoft Dynamics CRM 2011 Update Rollup 10 (UR10) and later, optimizing the performance of queries against large datasets involves adjusting specific “statistical” settings to address the issue, and should that fail to achieve desired levels of performance, to adjust the value associated with **EnableRetrieveMultipleOptimization** (ERMO) setting.

**Note**
Manual configuration of the ERMO setting overrides Microsoft Dynamics CRM’s ability to dynamically manage related settings. As a result, customers who have already configured the ERMO setting manually should remove the key entirely or reset the value to 0 (the default) to ensure that Microsoft Dynamics CRM can align to the updated default behavior that is included with UR10.

### Adjusting “statistical” settings

A first step in efforts to optimize the performance of queries against large data sets is to make adjustments to the “statistical” settings that affect the behavior of RetrieveMultiple queries:

- The **RecordCountLimitToSwitchToCteSecuritySql** setting (default value = 75,000) specifies the number of records below which Microsoft Dynamics CRM 2011 will default to an “OR”-based query using joins; when the number of records meets or exceeds this setting, Microsoft Dynamics CRM 2011 will consider a common table expression (CTE)-based query.

With a CTE-based query, SQL Server evaluates each Microsoft Dynamics CRM 2011 security filter independently within the query and earlier in its query optimization process, which can often result in better performing queries.

Consider using a higher value if performance is acceptable without requiring a CTE-based query for tables of larger size. Consider using a lower value if non-global-user performance is unacceptable for smaller-sized tables. Note that a lower number may have more of an impact on SQL Server memory use, so keeping the number as high as possible while still achieving acceptable levels of performance is preferred.

- The **RetrieveMultipleSharingCountThreshold** setting (default value = 1,000) specifies the number of shared records (for the entity type being retrieved) that are associated with a user directly or with the teams to which that user belongs. When the number of shares is below the value of this setting, Microsoft Dynamics CRM 2011 uses a table-valued function (TVF) to quickly locate records that are visible through sharing. When the number of shares meets or exceeds this setting, Microsoft Dynamics CRM 2011 will revert to using table joins to identify shared records rather than using a TVF to calculate sharing.

  Consider increasing this value if users with an above average number of shared records for an entity are experiencing a performance impact with that entity. However, for users with an extremely large number of shared records for an entity, taking advantage of the TVF for identifying shared records will likely be more expensive. In this case, if you are considering a higher value for this setting, be sure to increase the value in small (100 or 200) increments.

It is recommended that these settings be configured by using the OrgDbOrgSettings, which will ensure that the configuration applies to a specific organization.

**Note**

For additional information about configuring these settings by using the OrgDbOrgSettings, see the Knowledge Base article titled OrgDBOrgSettings Tool for Microsoft Dynamics CRM 2011

You can also configure these settings by modifying the Windows Registry (under HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSCRM) on the each of the Web servers that is used in a deployment, taking care to explicitly name each key name and data value. Note that configuring these settings via the Web server registry will apply the configuration to all organizations that use the Web servers.

**Warning**

Serious problems might occur if you modify the registry incorrectly by using Registry Editor or by using another method. These problems might require that you reinstall the operating system. Microsoft cannot guarantee that these problems can be solved. Modify the registry at your own risk.

**Configuring the EnableRetrieveMultipleOptimization setting**

With Microsoft Dynamics CRM 2011 UR10, security generation logic is dynamic and tries to build the appropriate query structure on a user-by-user and entity-by-entity basis. As a result, if making adjustments to these settings does not yield satisfactory levels of performance, then consider adjusting the value of the ERMO setting. The ERMO setting is commonly used to limit the

duration of certain long running queries (specifically, to overcome issues related to local or deep business unit access within Microsoft Dynamics CRM 2011 security roles).

The values that can be used with the ERMO setting are described in the following table:

| ERMO setting | Detail |
|---|---|
| 0 = Default | **Description**: Defaults to a CTE format for security filtering.<br>**When to consider**: Use in most cases – especially useful with larger data sets in which the number of visible records varies a lot between different users, because Microsoft Dynamics CRM 2011 uses the “statistical” settings called out previously to generate the most appropriate query structure for various users. Modifying the “statistical” settings can help tune how Microsoft Dynamics CRM 2011 generates queries to provide an optimal experience for each environment. |
| 1 = Use Temp Table | **Description**: The IDs of all records a user has Read access to are initially selected into a temporary table in SQL Server, and then the overall results of the user query are filtered against the IDs in the temporary table.<br>**When to consider**: Because SQL Server selects the IDs of all records a user has Read access to, consider using this value only when users typically have Read access to a small set of records – for example if a user can only see the records he or she owns and that number of records is relatively small (ideally less than a few thousand). The greater the number of IDs that SQL Server has to select into this temporary table, the slower the associated query will run. |
| 2 = Inline Security IDs | **Description**: Defaults to “OR”-based filtering for security, but the IDs of teams the user is a member of or business units the user has visibility into are passed in instead of being calculated through table joins in the query. |

| ERMO setting | Detail |
| --- | --- |
| | **When to consider**: Consider using when most users have Read access to most records without the use of sharing, and most users are members of a small number of teams and have access to records in a small number of business units. |
| 3 = OR Use Joins | **Description**: Defaults to “OR”-based filtering for security, and team membership and business unit visibility are always calculated through table joins. (Microsoft Dynamics CRM 4.0 used this type of security query structure.)<br>**When to consider**: For smaller organizations, this format of security filtering is often sufficient and may not require as much memory as the default ERMO setting. |

**Important**
Using non-default values for the ERMO setting can provide performance benefits in certain environments with specific customer data patterns; note that in some environments, using non-default values for the ERMO setting may impact performance negatively.

## Optimizing the performance of applications running against Microsoft Dynamics CRM 2011

Configuring applications running against Microsoft Dynamics CRM 2011 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.

#### Setting 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, customize default views to limit the records or accounts that are displayed.

**To set the default view for Accounts**

1. On the Microsoft Dynamics CRM home page, in the navigation bar, click **Settings**, under **Customization**, click **Customizations**, and then click **Customize the System**.
2. In the **Solution** window, under **Components**, expand **Entities**, expand **Account**, and then click **Views**.

   In the list of available views, the entry for the current default view is marked with a star. Its Type shows as **Default Public View**.
3. Select the alternate view you that you want to set as default, on the action toolbar click **More Actions**, and then click **Set Default**.
4. To save and publish the configuration changes, click **Publish All Customizations**.

**Customizing quick find views by limiting search columns**

When using the quick find feature for an entity, the results are displayed in the entity's Quick Find view. When customizing this view, you can define the columns returned, sorting and filter criteria, and the columns (fields) that will be searched.

The number of fields that are searched to display Quick Find results can directly affect performance. To ensure optimal performance, configure the Quick Find feature for an entity to search only the fields necessary to address specific business requirements.

**To customize Quick Find search parameters**

1. On the Microsoft Dynamics CRM home page, in the navigation bar, click **Settings**, under **Customization**, click **Customizations**, and then click **Customize the System**.
2. In the **Solution** window, under **Components**, expand **Entities**, expand the entity you want to customize the Quick Find view for, and then click **Views**.In the list of available views, select the **Quick Find** view, on the action toolbar click **More Actions**, and then click **Edit**.
3. In the **View: Quick Find** window, under **Common Tasks**, click **Add Find Columns**.
4. In the **Add Find Columns** dialog box, specify the fields that will be searched to provide Quick Find results, and then click **OK**.
5. In the **View: Quick Find** window, in the action bar, click **Save and Close**, and then in the **Solution** window, click **Publish All Customizations**.

When customizing Quick Find search parameters, if you experience slow query returns, leverage database optimization techniques and tools and be sure consider the following points:

- The more columns included in a search, the longer the search will take.
- Fields included in a search should be leading columns in indexes, even if this means creating one for each field. Also consider including in those indexes the Owner, BU, and the State fields, which are typically included in the query.
- Using filtered indexes can result in better query performance.

**Note**

For additional information about SQL indexes, see the “Optimizing and Maintaining the Microsoft Dynamics CRM Database” section of this document.

**Best Practices for Optimizing the Performance of Applications Running Against Microsoft Dynamics CRM 2011**

When optimizing the performance of applications running against Microsoft Dynamics CRM 2011, be sure to consider the following best practices.

1. **Leverage enhanced teams functionality.** In Microsoft Dynamics CRM 2011, teams can own objects and objects can be shared to teams allowing members of that team access to the shared objects. Users should consider leveraging teams instead of an excessively complex business hierarchy because teams will provide better performance with a lower penalty for security checks.
2. **Use Field Level Security (FLS) wisely.** FLS is a new feature available in Microsoft Dynamics CRM 2011. Using FLS provides a number of benefits in terms of providing more precise control over the data that specific users can access and view. However, there is a performance impact associated with using FLS, and the more extensively the feature is used in an implementation, the greater the impact on performance.

**Important**
Indexes added to Microsoft Dynamics CRM 4.0 database tables can be lost during an upgrade to Microsoft Dynamics CRM 2011. A recommended practice is to review any custom indexes in Microsoft Dynamics CRM 4.0 databases and then confirm the availability of those indexes after the upgrade.

### 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 the order of records returned by a saved query. There may be important business reasons associated with the order of display in query results. If there are business reasons for changing, consider adding an index based on the new ordering to improve the performance of the query.
- 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.

**Note**
For more information about using SQL Server indexes to enhance query performance in Microsoft Dynamics CRM 2011, see the white paper, Improving Microsoft Dynamics CRM Performance and Securing Data with Microsoft SQL Server 2008 R2.

### Disabling Auto-Complete on Lookups

Enabled by default, the Auto-Complete on Lookups feature in Microsoft Dynamics CRM 2011 can help to increase user efficiency, but it can also affect overall performance and resource usage in a Microsoft Dynamics CRM 2011 solution. To optimize performance, consider disabling this functionality.

**To disable Auto-Complete on Lookups**

1. Log on to Microsoft Dynamics CRM 2011 by using an account with the ability to Customize Entities.
2. Navigate to **Settings | Customize the System | Components | Entities | <entity> | Forms**, where <entity> represents the name of the entity to be modified, and then on the **<entity> Forms** page, double-click the form that includes the field you want to modify.
3. In Form Design view, select the field for which Auto-complete is configured, and then click **Change Properties**.
4. In the **Field Properties** window, on the **Display** tab, under **Field Behavior**, select the **Turn off automatic resolutions in the field** check box, and then click **OK**.
5. Repeat this procedure on each Entity/Field combination for which you want to disable the auto-complete feature.

**Querying on Custom Entities**

When you add custom attributes to Microsoft Dynamics CRM entities, the SQL Server database columns for those attributes 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. Creating the ideal index will likely be an iterative process, but implementing this correctly can yield significant performance benefits.

**Optimizing the performance of custom Microsoft Dynamics CRM SDK applications**

It is also important to ensure the optimal performance of any custom applications, plug-ins, or add-ins that have been developed by using the Microsoft Dynamics CRM 2011 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 in 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.

**Best Practices/Performance Characteristics for API-Based Applications**

The Microsoft Dynamics CRM API offers rich-client support for interacting with the Microsoft Dynamics CRM server (whether installed on premises or accessed via the cloud) and has been designed to minimize the amount of configuration required to make an SDK call. However, while integrating support for both WCF and WS-Trust into the API has allowed the presentation of a very similar authentication mechanism in all deployments, it has also increased the complexity on

the client side, both in discoverability of an SDK endpoint and in the creation of a client that can access it.

Fortunately, the complexity of these different configurations has been made largely transparent to end users through various helper classes available in the Microsoft.Xrm.Sdk.dll. This section describes expected performance characteristics and explains the best practices associated with using the SDK client.

**Warning**

To obtain the best performance from applications written to use the SDK, it is important to understand how the different components are designed to work together and potential factors that can impact performance. For more information about the components of the SKD and the interaction of those components, in the Microsoft Dynamics CRM 2011 SDK, see the topic Improve Service Channel Allocation Performance.

## Optimizing Microsoft Dynamics CRM Reporting Services

Report server performance can be affected by a variety 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 opt to deploy on a single server or on two servers, with one computer running Microsoft Dynamics CRM Server 2011 and the other computer running Microsoft SQL Server and SQL Server Reporting Services. With larger datasets, more users, or heavier loads, performance will be affected. If it is noted that the usage for reports has increased, optimizing reporting services would be required.

When optimizing Microsoft Dynamics CRM 2011 Reporting Services, consider these guidelines:

- Report processing and rendering are memory intensive operations, so ensure that the computer hosting the report server 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 with multiple report server instances. For best results, use load balancing software and hardware to distribute requests evenly across multiple report servers in the deployment. If load balancing is used, it is advisable to use client affinity to avoid loading the session cache in memory on multiple servers.
- 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, see the following resources:

- Microsoft Dynamics CRM Reporting Extensions.
- Planning for Scalability and Performance with Reporting Services.

- Using Visual Studio 2005 to Perform Load Testing on a SQL Server 2005 Reporting Services Report Server.

For more information, see Monitoring Report Server Performance.

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

# Optimizing report performance

The following sections provide guidelines and techniques for optimizing and maintaining the performance of reports.

## Optimization guidelines

End users want reports to come up quickly, and administrators need to limit the degree to which a single user's report activity impacts other users leveraging 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 data from a specified time frame, for example the previous 90 days, rather than displaying all records in the Microsoft Dynamics CRM database.
- Reports with a large dataset or a complex SQL query should not be available to all users on-demand. Instead, schedule a snapshot in Report Manager during a period when the system is lightly loaded.
- Deploy reports through Microsoft Dynamics CRM, and then use Report Manager to run the reports and have the results posted at a scheduled time.
- Reports should access the fewest datasets possible to meet business requirements.
- When possible, use fetch based reports, which are much more efficient than SQL-based reports, to run against Filtered Views.
- Consider using subreports. Show aggregated information in a report initially, and then use subreports for drilling down on aggregates to show non-aggregated values. Using this technique will prevent Report Server aggregations and improve performance.
- Use explicit paging for reports that require bringing large amounts of data to the Reporting Services middle tier. Build reports so that they show only a page of the total records at one time and have explicit clickable links to bring in data for further pages. Although Report Viewer control shows paging control on reports, all the data needed for the full report has already been pulled from the middle tier.

## 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 aggregating the data afterward in reporting services. This helps to prevent the computer running Microsoft SQL Server from being overtaxed with gathering, transmitting, and then processing large volumes of data. Instead, it uses the natural indexing and grouping ability of SQL Server to massively reduce this overhead.

### Making reports pre-filterable

When creating a report, you can make it “pre-filterable” by including a default filter for users to edit prior to running the report. Each user who customizes and runs a pre-filterable report effectively reduces the size of the data set and limits the amount of data pulled.

A key advantage of making a report pre-filterable is that the default filter selects active records that were modified in the last 30 days to prevent users from unintentionally running the report on all records. 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.

In Microsoft Dynamics CRM 2011, to make a SQL-based report pre-filterable, specify the CRMAF_ prefix in the SQL query during the creation of the report in Report Designer. Adding this prefix to at least one filtered view in the query ensures that 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.

**Warning**

For information about parameterization of fetch-based reports in Microsoft Dynamics CRM 2011, on the Microsoft Dynamics CRM Blog, see the article Parameterizing Fetch Based Reports.

### 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 the report more restrictive. If a field in the WHERE clause is used frequently, verify that a non-clustered index exists on that field.

### Throttling resources used for reports and data visualizations

A server running Microsoft Dynamics CRM Server 2011 (Application tier) tags the queries it sends to SQL Server (Data tier) to designate the purpose of the queries.

If a query is sent for reporting purposes, it is tagged with MSCRMReportsGroup. If it is sent for data visualization, it is tagged with MSCRMVisualizationsGroup. As a result, queries can be throttled by the SQL Resource Governor based on source, allowing administrators to control the maximum SQL Server system resources used for CRM Reports and CRM Data Visualizations.

**Note**

For more information on SQL Server Resource Governor and how to use it to manage SQL Server resource consumption, see the following resources:

- Resource Governor.
- Using the Resource Governor.
- ALTER RESOURCE GOVERNOR (Transact-SQL).

## Best practices for optimizing workflow

Enabling new workflow options in Microsoft Dynamics CRM 2011 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 goals of workflow within the solution. Microsoft Dynamics CRM 2011 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 troubleshoot 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.
- Account for the overall load associated with workflow within a deployment. Consider the number of instances that each workflow definition triggers and the load of workflows (number of workflows, which entities, number of records, data size, and data load) in the system on a typical day to better understand the processes and load variances, and, based on this analysis, optimize the workflows as needed.
- Review workflow logic carefully. Workflows that include infinite loopbacks, which, because of semantic or logic errors, can never terminate through normal means, can greatly affect overall workflow performance. When creating and implementing workflow functionality within a Microsoft Dynamics CRM 2011 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 that 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.

- To improve performance in large deployments, scale out as necessary. For large-scale deployments, use dedicated computers to run the Async service. Keep in mind, however, that increasing the number of servers running the Async service creates additional stress on the server running Microsoft SQL Server. As a result, be sure to follow appropriate optimization and tuning strategies on the data tier and investigate the possibility of increasing the number of computers running Microsoft SQL server.
- Test workflows. Be sure to test and monitor the performance of new workflow functionality before implementing it in a production environment.
- Async plug-ins. Consider whether plug-ins should run synchronously or asynchronously. If the priority is user responsiveness, running a plug-in asynchronously will enable the user interface to respond more quickly to the user. However, asynchronous plug-ins introduce additional load to the server to persist the asynchronous operation to the database and process by the Async Service. If scalability is more important, running plug-ins synchronously typically requires less load on the servers than running plug-ins asynchronously.
- Balancing asynchronous plug-ins and workflows. Asynchronous plug-in and workflow records in the asyncoperationbase table are managed with equal priority. Therefore, introducing a large number of asynchronous plug-ins into a system can reduce overall throughput or increase the time between triggering and processing of individual workflows. As a result, be sure to consider the relative importance of the workflows in the system before adding numerous asynchronous plug-ins to the solution.
- Child Workflows. Child workflows run as independent workflow instances from their parents. This can enable parallel processing on a system with spare capacity, which can be useful for workflows with multiple independent threads of high processing activity. It can also introduce additional overhead if the parallel processing is not critical because other workflow logic threads are blocked waiting for external events to occur.

  **Warning**

  If the existing workflow functionality within a Microsoft Dynamics CRM 2011 implementation is not performing as expected, verify that the Async service is running properly. Often, restarting the Async service will unblock workflow processing without affecting the functionality of the workflows that were in the pipeline.
- Monitor the Microsoft Dynamics CRM 2011 database for excess workflow log records. Workflow processing in Microsoft Dynamics CRM relies on the Asynchronous Service, which logs its activity in both the AsyncOperationBase table and WorkflowLogBase tables. As the number of workflow records in the Microsoft Dynamics CRM 2011 database grows over time, performance may be affected.

Microsoft Dynamics CRM 2011 includes two specific settings, **AsyncRemoveCompletedJobs** and **AsyncRemoveCompletedWorkflows**, which can be configured to ensure that Asynchronous Service automatically removes log entries from the AsyncOperationBase and WorkflowLogBase tables.

- The **AsyncRemoveCompletedWorkflows** setting is exposed to users in the interface for defining new workflows, and users can set the Removal flag independently on each of the workflows they define.

  **Note**

When registering an Async plug-in, users can also specify that successfully completed Async Plugin execution records be deleted from the system.

- Users can change the **AsyncRemoveCompletedJobs** setting by using the deployment Web service. However, the setting is by default configured to **True**, which ensures automatic removal of entries for successfully completed jobs from the AsyncOperationBase table.

**Important**
When you install or upgrade to Microsoft Dynamics CRM 2011, a recurring "BulkDelete" async operation will be created by setup. This operation runs every day to identify any remaining async operations that were completed more than 30 days ago. Depending on business requirements, you can customize this bulk delete operation or remove it completely.

**Note**
For more information about removing workflow log records from the Microsoft Dynamics CRM database, in the Microsoft Dynamics CRM 2011 SDK, see the following resources:

- Delete Data in Bulk in Microsoft Dynamics CRM.
- Run Bulk Delete.
- BulkDeleteOperation Entity Messages and Methods.
- BulkDeleteRequest Class.

**Important**
The Workflow Engine includes an internal system setting that specifies the interval at which jobs are polled for processing. If you have a strong business case for modifying this setting, contact Microsoft Dynamics CRM 2011 support. For more information about Microsoft Dynamics CRM 2011 support options, see Help on Hot Topics for New Users.

**Note**
For additional information about best practices and guidelines associated with workflow design, on the Microsoft Dynamics CRM Blog, see the entry titled Microsoft Dynamics CRM: Workflow Authoring Best Practices.

# Optimizing and maintaining the data tier

The following sections provide detail about techniques associated with optimizing and maintaining the data tier.

## Optimizing and maintaining Microsoft SQL Server

Microsoft SQL Server performance can be affected by a variety of factors, from poor database design to improper server configuration. To help identify and resolve issues that can affect SQL Server performance, there are a variety of publicly available tools such as:

- SQL Server Extended Events (XEvents) and the data collector, which are new in SQL Server 2008

- SQL Server Profiler
- Dynamic management views (sometimes referred to as DMVs)
- Performance Monitor

However, as with other components in a Microsoft Dynamics CRM 2011 implementation, before attempting to optimize the performance of Microsoft SQL Server, be sure to verify that the computer running Microsoft SQL Server includes the appropriate hardware based on usage within a specific environment.

**Warning**

For a listing of the hardware requirements for running Microsoft SQL Server, in the Implementation Guide, in the Planning Guide, see the topic Microsoft SQL Server hardware requirements.

In addition to performing the standard recommended practices for maintaining Microsoft SQL Server, be sure 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 optimizing and maintaining the performance of Microsoft SQL Server, see the following resources:

- Troubleshooting Performance Problems in SQL Server 2008.
- Microsoft® SQL Server® 2008 R2 Best Practices Analyzer.
- Microsoft SQL Server TechCenter.
- Microsoft SQL Server Performance Blog.

**Important**

Microsoft SQL Server 2008 features, including Compression, Sparse Columns, and Backup Compression, when implemented properly, can improve the performance of a Microsoft Dynamics CRM implementation.

For more information about improving Microsoft Dynamics CRM performance with Microsoft SQL Server 2008, see the white paper Improving Microsoft Dynamics CRM Performance and Securing Data with Microsoft SQL Server 2008 R2.

## Partition alignment in Windows operating systems

Disk partition alignment is a powerful technique for improving the performance of Microsoft SQL Server. Configuring optimal disk performance is often viewed as much art as science. An essential yet often overlooked best practice is to properly align disk partitions. While Windows Server 2008 attempts to align new partitions out-of-the-box, disk partition alignment remains a relevant technology for partitions created on earlier versions of Windows.

**Note**

For more information about partition alignment for optional performance, on TechNet, see the article Disk Partition Alignment Best Practices for SQL Server.

Partition alignment works differently depending on the version of Windows being used and the version in which the partition alignment was created.

### Windows Server 2008: New partitions

In Windows Server 2008 (and Windows Vista), partition alignment is usually performed by default. The default for disks larger than 4 GB is 1 MB; the setting is configurable and is found in the registry at the following location:

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\VDS\Alignment

However, even computers pre-installed with Windows Server 2008 can have partitions with undesirable partition starting offsets, so regardless of the operating system, be sure to confirm that new partitions are properly aligned.

### Windows Server 2008: Pre-existing partitions

New partitions on Windows Server 2008 are likely to be aligned. However, partitions created on earlier versions of Windows that become associated with Windows Server 2008 will not be aligned unless partition alignment is explicitly performed on those partitions.

## Disabling support for parallel plan queries

On computers with multiple processors, 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. SQL Server considers parallel execution plans for queries, index data definition language (DDL) operations, and static keyset-driven cursor population.

In a default configuration of Microsoft SQL server, the “max degree of parallelism” value is set to 0, which specifies to use all available processors up to a maximum of 64. Setting this value to 1 will suppress all parallel plan generation, while setting the value to a number greater than 1 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 by using the sp_configure system stored procedure.

**Important**

“Max degree of parallelism” is an advanced setting that should be changed only by an experienced database administrator or certified SQL Server technician. To use the sp_configure system stored procedure to change the setting, the “show advanced options” must be set to 1. The setting takes effect immediately (without restarting the MSSQLSERVER service). For more information, see the topic max degree of parallelism Option.

For more information about setting the “max degree of parallelism” value in SQL Server Management Studio, see Configure the max degree of parallelism Server Configuration Option.

Note that you can override the max degree of parallelism value in queries by specifying the MAXDOP query hint in the query statement. For more information, see Query Hints (Transact-SQL).

In addition, index operations that create or rebuild an index, or that drop a clustered index, can be resource intensive. You can override the max degree of parallelism value for index operations by specifying the MAXDOP index option in the index statement. The MAXDOP value is applied to the statement at execution time and is not stored in the index metadata. For more information, see Configure Parallel Index Operations.

## Using efficient queries

When planning a query strategy, keep in mind these guidelines for designing efficient queries.

- Retrieve only the data required to meet a specific business requirement. Retrieving more data than required can lead to increased network traffic and require additional server and client resources.
- Avoid mixing OLTP, OLAP, and reporting workloads. OLTP workloads typically have many small transactions, with very quick response time expected from the user. OLAP and reporting workloads typically have a few long-running operations that can 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 de-normalization) 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.

**Note**

For more information about using indexes properly, see the section of this paper titled Optimizing and Maintaining Database Indexes.

## Optimizing and maintaining query performance

Based upon the types of queries that are run frequently and the underlying hardware configuration, you can optimize query performance by modifying the way the application displays query results; by partitioning a table or index; or by using multiple disk drives. Also make sure to maintain queries on a regular basis to ensure optimal performance.

For additional information about partitioning tables and indexes, and optimizing SQL query performance, in SQL Server Books Online, see the topic Designing Partitions to Improve Query Performance.

## Optimizing the performance of Quick Find queries

Ensuring that Quick Find queries perform optimally requires defining the column or set of columns to be used in a Quick Find query, and then ensuring that a non-clustered index is in place for each of those columns. When creating Quick Find searches with performance in mind, be sure to consider the following best practices and recommendations.

**Important**

With Microsoft Dynamics CRM 2011 Update Rollup 10 (UR10), the implementation of Quick Find functionality has been updated to better take advantage of indexes and the SQL query engine to provide better performance. The new Quick Find query structure can be disabled with an OrgDbOrgSetting - specifically by setting the value of the **EnableQuickFindOptimization** setting to 0.

### Best practices for Quick Find queries

When considering the need to optimize the performance of Quick Find queries, be sure to keep in mind the following best practices:

- Limit the number of columns selected only to those necessary to meet business requirements. Quick Find queries that include more than eight-to-ten well-suited columns (as defined below) may adversely impact performance.
- Quick Find functionality is optimized for locating one record or a small set of records. As a result, it is recommended to include search fields that mostly include unique values across records. Searching on phone numbers, email addresses, names, or other unique values (including custom attributes that are likewise well suited for Quick Find searches) will yield satisfactory results. On the other hand, searching on address data (e.g. zip codes, city name, and so on), while supported, is discouraged because values in these field tend to be common across multiple records.
- Avoid searching on picklist attributes; this would result in a search of label that can be localized, and these types of columns currently cannot be indexed.
- Avoid search terms that begin with a wildcard character (e.g. first character is '*') – queries using these types of search terms cannot use indexes to optimize performance.
- Avoid using search terms that are not very selective.

In addition, Microsoft Dynamics CRM 2011 UR10 introduces a new option in System Settings, **Select whether quick find record limits are enabled**, which can help to improve performance by protecting the system from runaway queries. For example, with this setting enabled, if a user performs a Quick Find search that returns 10,000 or more records, rather than presenting the user with the search results, the system will alert the user with the following message:

*Quick find limit exceeded. Please use a more selective search value, or use Advanced Find for your search.*

If users need to search using non-selective terms that will potentially match over 10,000 records, this setting can be disabled. However, this is not recommended because disabling the setting may result in slower overall performance for Quick Find queries. These types of queries are best served with Advanced Find.

**Warning**

Information for Developers about SDK changes that support UR10 Quick Find query performance improvements is available in the following Microsoft Dynamics CRM SDK topic: Build Queries with QueryExpression.

**Important**

Outlook clients that are Microsoft Dynamics CRM 2011 and Microsoft Dynamics CRM Online UR 6 or earlier have a different implementation of Quick Find that performs less optimally than does the Web client. Beginning with UR7, a new implementation of Quick Find yields levels of performance similar to those experienced with the Web client.

### Index recommendations and requirements

In addition to the best practices associated with developing Quick Find queries in Microsoft Dynamics CRM 2011, be sure to consider the following recommendations and requirements related to indexes:

- Create an index for each search field that is used in a Quick Find query. Each search field should be the leading column in an index.
- For searches against data in an entity's base table or extension table, which is physically stored with the entity definition, create one index per column, ensuring that the column is the leading column in the index key.
- For searches against address data, which is stored in a separate table, create an index for each search field on the entity's address table with an index key of the form (<**SearchField**>, **AddressNumber**). In addition, the index should add the **ParentId** column as an include column. For example, if searching on a Lead's *address2_city* field, the index would be of the form:

```
CREATE NONCLUSTERED INDEX [ndx_LeadAddress_City] ON
[dbo].[LeadAddressBase]
(
[City] ASC,
[AddressNumber] ASC
)
INCLUDE ( [ParentId])
```

  Addresses for the Account and Contact entities are both stored in the CustomerAddressBase table. This table includes an ObjectTypeCode column to delineate addresses belonging to Accounts from addresses belonging to Contacts. As such, the index key must also include the ObjectTypeCode column in the form (<**SearchField**>, **AddressNumber**,

**ObjectTypeCode**). For example, if searching on an Account's *address1_line1* field, the index would be of the form:

```
CREATE NONCLUSTERED INDEX [ndx_AddressLine1] ON
[dbo].[CustomerAddressBase]
(
[Line1] ASC,
[AddressNumber] ASC,
[ObjectTypeCode] ASC
)
INCLUDE ( [ParentId])
```

- For searches against related-entity names that are physically stored on the related-entity base or extension table, create one index on the foreign key ID field of the Quick Find entity and another index on the primary name field of the related entity, if one is not already present.

**Important**
Administrators cannot currently define indexes against a Microsoft Dynamics CRM Online organization.

**Note**
For additional information about SQL indexes, see the "Optimizing and Maintaining the Microsoft Dynamics CRM Database" section of this document.

## Enhancing the performance of the display of query results

By default, when a user runs a query in Microsoft Dynamics CRM, the results are displayed in sets of 5000 records across multiple pages. As a result, Microsoft Dynamics CRM initially displays the first 5001 records, then the first 10,001 records, then the first 15,001 records, and so on. So, while the query is relatively "cheap" initially, the expense grows exponentially with each subsequent display.

For better performance when you retrieve a subsequent "page" of results, assign the value from the PageInfo property (returned with the collection of query results) to the PagingInfo.PagingCookie property in the QueryExpression class or the QueryByAttribute class

**Note**
For additional information about enhancing the performance of the display of query results, in the Microsoft Dynamics CRM 2011 SDK, see the topic Using the Paging Cookie (PageInfo).

### 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 collocated so 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 collocated or are not partitioned on the join field, the presence of partitions may actually slow down query processing instead of accelerate it.

**Important**

When partitioning for join queries, be sure to verify the performance gains associated with any modification you make. Also keep in mind that partitions created for join queries will have to be recreated in case of an upgrade.

### 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 and other aspects in the database, including indexes and plan guides, to enhance performance and manageability of databases.

**Warning**

For more information about optimizing and maintaining the Microsoft Dynamics CRM database, see Optimizing Workflow in this document. Also, in SQL Server Books Online, see Designing and Creating Databases.

## Segregating the database, tempdb, and transaction log files

Creation of transaction log files 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, be sure that database files, the tempdb files, and transaction log files are each located on separate sets of physical disks.

## Optimizing and maintaining database indexes

Optimizing and maintaining database indexes is a critical step in managing the overall performance of the Microsoft Dynamics CRM 2011 database. In fact, 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. For large databases, consider creating indexes on separate file groups.

Keep in mind that adding indexes comes with an additional cost. As more indexes are added, inserting and updating data in the database table becomes slower because of the changes required on the indexed data. Additionally, more indexes in a database increase the database size, thus increasing maintenance time and the time it takes to back up the database.

**Note**

For more information, in SQL Server Books Online, see the following resources:

- Optimizing Indexes.
- ALTER INDEX (Transact-SQL).

**Important**

While creating customized indexes can help to improve query performance within a Microsoft Dynamics CRM implementation, be sure to avoid making any modifications to the default indexes that are included with Microsoft Dynamics CRM 2011.

### Rebuilding Indexes

Rebuilding an index drops and re-creates the index, which 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. Specifying ALL ensures that 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.

**Online Index Maintenance**

You can create, rebuild, or drop indexes online. The ONLINE option allows concurrent user access to the underlying table or clustered index data and any associated nonclustered indexes during these index operations. For more information, see Performing Index Operations Online.

**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.

## Implementing Solid State Drive technology

Database I/O performance is a key factor for determining Microsoft Dynamics CRM 2011 application scalability. Upgrading your storage subsystem will yield better performance because of higher IOPS. Solid State Drives (SSDs) provide high IOPs and are well suited for the Microsoft Dynamics CRM application I/O profile, which tends to be random access to small pieces of data.

SSD technology typically provides faster system performance than the magnetic media associated with traditional hard disk drives (HDDs). While more expensive than HDDs, using SSDs in a system can also enhance overall system responsiveness, and they generally consume less power.

Customers can also choose to use SSDs only for a subset of the Microsoft Dynamics CRM database. Based on frequency of access, partition the database into filegroups so the most frequently accesses tables reside on SSDs, while the remainder continues to reside on HDDs.

## Implementing SQL virtualization

Based on hypervisor technology, the Hyper-V virtualization feature in the Windows Server 2008 operating system is a thin layer of software between the hardware and the operating system that allows multiple operating systems to run, unmodified, on a host computer at the same time. Hyper-V is a powerful virtualization technology that corporate IT departments can use to consolidate under-utilized servers, lowering total cost of ownership (TCO) and maintaining or improving quality of service (QoS).

Regarding performance, Hyper-V is a viable option for SQL Server consolidation scenarios, as the overall performance of SQL Server running in a Hyper-V virtualized environment is reasonable compared to an equivalent native Windows Server 2008 environment.

With proper I/O capacity and configuration, the I/O overhead is minimal. For best performance, you should have enough physical processors to support the number of virtual processors configured on the server to avoid overcommitting CPU resources. The CPU overhead increases significantly when the CPU resources are overcommitted. It is important to test each application thoroughly before you deploy it to a Hyper-V environment in production.

**Important**

For additional information about implementing SQL Virtualization and some general considerations and recommendations when running SQL Server in Hyper-V environments, see the white paper Running SQL Server 2008 in a Hyper-V Environment: Best Practices and Performance Considerations.

Also be sure to consider the following resources:

- Optimizing SQL Server for Private Cloud.
- Running SQL Server 2008 in a Hyper-V Environment - Best Practices and Performance Recommendations.

# Optimizing and maintaining the Microsoft Dynamics CRM Email Router

## Optimizing the Microsoft Dynamics CRM Email Router

To help ensure that the Microsoft Dynamics CRM Email Router performs optimally within a specific Microsoft Dynamics CRM solution, keep the following points in mind:

- Ensure that the computer designated to run the Email Router is configured with the appropriate hardware and software based on usage.

  **Note**

  For a listing of the hardware and software requirements for running the Email Router, in the Implementation Guide, in the Planning Guide, see the following topics:

  - Microsoft Dynamics CRM Email Router hardware requirements.
  - Microsoft Dynamics CRM Email Router software requirements.
- Configure the Email 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.
- Increase the number of Email Routers in a Microsoft Dynamics CRM 2011 deployment to provide better throughput. Performance will scale linearly up to a point, at which adding Email Routers will yield only modest gains. Because the number of servers appropriate for any specific deployment is dependent on a variety of environmental factors, be sure to monitor throughput as servers are added to determine when the point of diminished returns has been reached.

  In case of heavy email traffic, also be sure to consider:

  - Separating the inbound role and outbound role on different servers.
  - Using groups to distribute users between the multiple routers.
- For inactive or invalid users and queues, update the “incoming-delivery-method” from the Email Router to **None** to help reduce incoming and outgoing polling.

In addition, be sure to customize the default configuration of the Email 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 Email Router from “hanging” on a particular mailbox. |
| **Maximum Messages Per Cycle** (default = 1000) | When configured to process multiple forward mailboxes, the Email Router uses parallel processing to act on two mailboxes simultaneously. However, a single forward mailbox receiving a sudden blast of emails 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 emails the Email Router would process in a single mailbox before automatically freeing up a thread to advance to the next forward mailbox. |
| **Polling Period (seconds)** (default = 60) | This setting controls the duration of the sleep period between provider cycles. In general, increasing the duration of the sleep period decreases the need for system resources. However, if the Email Router must process several mailboxes, a lengthy 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 email before moved it to the undeliverable folder. Note that if the value is set too low, a simple Microsoft Dynamics CRM server restart may cause failures, while if the value is set too high, excessive retries may waste system resources. |

### Maintaining the Microsoft Dynamics CRM Email Router

To maintain the performance of the Microsoft Dynamics CRM Email 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 emails, for example, because of a spam attack, or because the email router was shut down.
- For verbose logging, set the Email Router service registry key “LogLevel” (located under HKYE_LOCAL_MACHINE\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**

For a complete listing of the performance counters available for tracking email processing and managing the overall performance of the Email Router in a Microsoft Dynamics CRM 2011 implementation, on the Microsoft Download Center, see the white paper Microsoft Dynamics CRM 2011 Performance Counters.

# Appendix A: Additional resources

## Technical resources

For more information about optimizing and maintaining Microsoft Dynamics CRM 2011, see the following technical resources.

**Warning**

For the latest information about Microsoft Dynamics CRM performance in general and to learn about the release of additional support resources, see the Microsoft Dynamics CRM Team Blog or the Microsoft Dynamics CRM Online Team Blog.

## Implementation and installation

- Microsoft Dynamics CRM 2011 Implementation Guide
  - Download
  - View online
- Microsoft Dynamics CRM 2011 for Outlook Installing Guide for use with Microsoft Dynamics CRM Online
- Microsoft Dynamics CRM 2011 Readme

## Optimization and maintenance

- Optimizing and Maintaining Client Performance for Microsoft Dynamics CRM 2011 and CRM Online:
- Performance Toolkit for Microsoft Dynamics CRM 2011
- Bandwidth testing using Microsoft Dynamics CRM Performance Toolkit
- Improving Microsoft Dynamics CRM Performance and Securing Data with Microsoft SQL Server 2008 R2
- SAMPLE - Performance and Scalability Assessment of Customer Implementation
- Microsoft Dynamics CRM 2011 performance counters

## "Nuts and bolts"

- Outlook Synchronization in Microsoft Dynamics CRM*
  * Microsoft Dynamics CRM4.0 version prior to update for Microsoft Dynamics CRM 2013
- Offline and Online Synchronization in Microsoft Dynamics CRM*
  * Microsoft Dynamics CRM4.0 version prior to update for Microsoft Dynamics CRM 2013

## Development

- Microsoft Dynamics CRM 2011 Software Development Kit
- Microsoft Dynamics CRM Developers Center

## Microsoft Services

Microsoft Services is the consulting and enterprise support division of Microsoft. Their mission is to help businesses around the world get a maximized return on their investment in Microsoft products and technologies. This means not only helping with deploying and optimizing IT, but also helping businesses move forward with IT initiatives that deliver the most business value.

Between Enterprise Services and Customer Support Services, Microsoft Services has a global team of more than 9,720 professionals who work with more than 60,000 partners across 88 countries. As a result, Microsoft Services can extend its services to take on even the largest projects, helping a large number of customers worldwide get the most out of existing IT assets, saving them money, and delivering real business results.

Microsoft Services supports customers through a variety of services:

- Strategy
  - Enterprise Strategy Portfolio
  - Enterprise Strategy Foundation
- Support
  - Premier Support
  - Microsoft Services Partner Advantage
  - Contact Microsoft Services
- Consulting

- Business Intelligence
- CRM Solutions
- Datacenter Services
- Dynamic Workplace
- Next Generation Applications
- Optimized Desktop
- Platform Modernization

**Note**
For more information about Microsoft Services, see the Microsoft Services web site.

# Appendix B: Accessibility for Microsoft Dynamics CRM

Administrators and users who have administrative responsibilities typically use the Settings area of the Microsoft Dynamics CRM web application to manage Microsoft Dynamics CRM. A mouse and keyboard are the typical devices that administrators use to interact with the application.

Users who don't use a mouse can use a keyboard to navigate the user interface and complete actions. The ability to use the keyboard in this way is a result of support for keyboard interactions that a browser provides.

For more information, see the following Microsoft Dynamics CRM web application accessibility topics:

- Use Keyboard Shortcuts
- Accessibility for People with Disabilities

Administrators and users who have administrative responsibilities for on-premises deployments of Microsoft Dynamics CRM 2011 also use Microsoft Dynamics CRM Deployment Manager, a Microsoft Management Console (MMC) application, to manage on-premises deployments of Microsoft Dynamics CRM Server 2011.

For more information, see the following Microsoft Management Console (MMC) accessibility topics:

- Navigation in MMC Using the Keyboard and Mouse
- MMC Keyboard Shortcuts

**Accessibility Features in Browsers**

| Browser | Documentation |
|---|---|
| Internet Explorer | Microsoft Accessibility<br>Language Support and Accessibility Features |
| Mozilla Firefox | Accessibility features in Firefox |

| Browser | Documentation |
| --- | --- |
| Apple Safari | Safari |
| Google Chrome | Accessibility Technical Documentation |

**Note**
For additional information, see the Microsoft Accessibility Resource Center

# Feedback

We appreciate hearing from you. To send your feedback, click the link below and type your comments in the message body.

**Note**
The subject-line information is used to route your feedback. If you remove or modify the subject line, we may be unable to process your feedback.

Send feedback

# Conclusion