# Registry Settings that can be Modified to Improve Operating System Performance



This section provides a description of recommended values for several registry entries that impact operating system performance. These registry entries can be applied manually or can be applied via the operating system optimization PowerShell script included in Windows PowerShell Scripts.

| Important |
| --- |
| During performance testing completed for this guide it was observed that Windows Server 2008 appears to be tuned by default such that modification of these registry entries did not provide the same performance benefits that have been observed on Windows Server 2003. Modification of these registry settings should only be done after a careful analysis of the effects on the system. |

## Registry settings that can be modified to improve operating system performance

### Increase available worker threads

At system startup, Windows creates several server threads that operate as part of the System process. These are called *system worker threads*. They exist with the sole purpose of performing work on the behalf of other threads generated by the kernel, system device drivers, the system executive and other components. When one of these components puts a work item in a queue, a thread is assigned to process it.
The number of system worker threads should ideally be high enough to accept work tasks as soon as they become assigned. The trade off, of course, is that worker threads sitting idle consume system resources unnecessarily. Modify and/or create the following REG_DWORD values in the registry and then set to the recommended values listed below.

recommended values listed below.

The **AdditionalDelayedWorkerThreads** value increases the number of delayed worker threads created for the specified work queue. Delayed worker threads process work items that are not considered time-critical and can have their memory stack paged out while waiting for work items. An insufficient number of threads will reduce the rate at which work items are serviced; a value that is too high will consume system resources unnecessarily.

## AdditionalDelayedWorkerThreads

| | |
|---|---|
| Key: | HKLM\SYSTEM\CurrentControlSet\Control\SessionManager\Executive |
| Value: | AdditionalDelayedWorkerThreads |
| Data Type: | REG_DWORD |
| Range: | 0x0 (default) to 0x10 (16) |
| Recommended value: | 0x10 (16) |
| Value exists by default? | **Yes** |

The **AdditionalCriticalWorkerThreads** value increases the number of critical worker threads created for a specified work queue. Critical worker threads process time-critical work items and have their stack present in physical memory at all times. An insufficient number of threads will reduce the rate at which time-critical work items are serviced; a value that is too high will consume system resources unnecessarily.

## AdditionalCriticalWorkerThreads

| | |
|---|---|
| Key: | HKLM\SYSTEM\CurrentControlSet\Control\SessionManager\Executive |
| Value: | AdditionalCriticalWorkerThreads |
| Data Type: | REG_DWORD |
| Range: | 0x0 (default) to 0x10 (16) |

| | |
|---|---|
| Recommended value: | 0x10 (16) |
| Value exists by default? | **Yes** |

## Disable Windows Server 2003 SP 1 and SP2 denial of service checking

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

| **Important** |
|---|
| Only disable this feature in an intranet scenario when you are sure you will not suffer from actual denial of service attacks. |

For more information about disabling Windows Server Denial of Service Checking, see Microsoft Knowledge Base article 889599, "A BizTalk Server Host instance fails, and a 'General Network' error is written to the Application log when the BizTalk Server-based server processes a high volume of documents" (http://go.microsoft.com/fwlink/?LinkID=153332). Follow the instructions in this article to create the **SynAttackProtect** registry entry on computers running SQL Server that host BizTalk Server databases and on any computers running BizTalk Server running Windows Server 2003 SP1 or later.

**Registry settings that govern the level of denial of service attack protection on Windows Server 2003** - In certain scenarios you may want to maintain denial of service protection but reduce how aggressively the denial of service functionality is applied. It is possible to tune the default behavior of the denial of service protection feature by following these steps:

1. Ensure the **SynAttackProtect** registry entry is set to a REG_DWORD value of **1** as described at SynAttackProtect (http://go.microsoft.com/fwlink/?LinkId=111477).

2. Configure the **TcpMaxHalfOpen** registry entry as described at TcpMaxHalfOpen (http://go.microsoft.com/fwlink/?LinkId=111478).

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

**Important**

The **SynAttackProtect**, **TcpMaxHalfOpen**, and **TcpMaxHalfOpenRetried** registry entries are no longer used with Windows Vista and Windows Server 2008. The TCP/IP protocol suite implementation in Windows Vista and Windows Server 2008 was redesigned to provide improved performance and does not require manual modification of these registry entries.

## Disable LDAP client signing requirements for computers in a secure intranet environment

Computers running Windows XP with Service Pack 1 (SP1) and higher, and computers running Windows Server 2003 SP3 and higher provide the ability to enforce LDAP client signing requirements to mitigate "man in the middle" attacks where an intruder captures packets between a client and a server, modifies them, and then forwards them to the server. When this occurs on an LDAP server, an attacker could cause a server to respond based on false queries from the LDAP client. Such "man-in-the-middle" attacks can be mitigated by requiring digital signatures on all network packets by means of IPSec authentication headers. Computers in a secure intranet environment should disable this option to reduce the overhead associated with LDAP client signing. You can modify this setting on computers running Windows Server 2003 SP3 and higher by changing the following registry entry from a REG_DWORD value of 1 to a REG_DWORD value of 0:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ldap\ldapclientintegrity = REG_D
```

## Increase space available for the master file table

Add the **NtfsMftZoneReservation** entry to the registry to allow the master file table (MFT) to grow optimally. When you add this entry to the registry, the system reserves space on the volume for the master file table. If your NTFS volumes contain relatively few large files, set the value of this registry entry to 1 (the default). Typically you can set this entry to a value of 2 or 3 for volumes that contain a moderate numbers of files, and use a value of 4 (the maximum) if your volumes tend to contain a relatively large number of files.

**Important**

Test any settings greater than 2, because setting this entry to a value greater than 2 will cause the system

to reserve a much larger portion of the disk for the master file table.

### NtfsMftZoneReservation

| | |
|---|---|
| Key: | HKLM\SYSTEM\CurrentControlSet\Control\FileSystem |
| Value: | NtfsMftZoneReservation |
| Data Type: | REG_DWORD |
| Range: | 1 – 4 |
| Default value: | 1 |
| Recommended value: | • 1 if volumes typically store fewer files.<br>• 2 or 3 if volumes typically store a moderate number of files.<br>• 4 if volumes typically store a large number of files. |
| Value exists by default? | **No**, needs to be added. |

## Change registry settings to maximize server service performance

The maximum number of concurrent outstanding network requests between a Windows Server Message Block (SMB) client and server is determined when a session between the client and server is negotiated. The maximum value negotiated is determined by registry settings on both the client and server. If these values are set too low on the server, they can restrict the number of client sessions that can be established with the server.

The values that can be adjusted to improve system performance for work items exist in the **LanmanServer** and **LanmanWorkstation** registry keys and are:

- MaxWorkItems

- MaxMpxCt
- MaxCmds

| Note |
| --- |
| These values do not exist in the registry by default. |

The **MaxWorkItems** value specifies the maximum number of receive buffers, or work items, the Server service is permitted to allocate at one time. If this limit is reached, then the transport must initiate flow control, which can significantly reduce performance.

## MaxWorkItems

| | |
| --- | --- |
| Key: | HKLM\SYSTEM\CurrentControlSet\Services\LanmanServer\Parameters |
| Value: | MaxWorkItems |
| Data Type: | REG_DWORD |
| Range: | 0 – 65535 |
| Default value: | Configured dynamically |
| Recommended value: | 8192<br>Note<br>The MaxWorkItems value must be at least four times as large as the MaxMpxCt value. |
| Value exists by default? | **No**, needs to be added. |

The **MaxMpxCt** value enforces the maximum number of simultaneous outstanding requests from a particular

client to a server. During negotiation of a Server Message Block between the client and the server, this value is passed to the client's redirector where the limit on outstanding requests is enforced. A higher value can increase server performance but requires more use of server work items (MaxWorkItems).

## MaxMpxCt

| | |
|---|---|
| Key: | HKLM\SYSTEM\CurrentControlSet\Services\LanmanServer\Parameters |
| Value: | MaxMpxCt |
| Data Type: | REG_DWORD |
| Range: | 0 – 65535 |
| Default value: | 50 |
| Recommended value: | 2048<br>Note<br>The MaxWorkItems value must be at least four times as large as the MaxMpxCt value. |
| Value exists by default? | **No**, needs to be added. |

The **MaxCmds** value specifies the maximum number of network control blocks the redirector can reserve. The value of this entry coincides with the number of execution threads that can be outstanding simultaneously. Increasing this value will improve network throughput, especially if you are running applications that perform more than 15 operations simultaneously. This value is set on the SMB client computer.

## MaxCmds

| | |
|---|---|
| Key: | HKLM\SYSTEM\CurrentControlSet\Services\LanmanWorkstation\Parameters |
| Value: | MaxCmds |

| | |
|---|---|
| Data Type: | REG_DWORD |
| Range: | 0 – 65535 |
| Default value: | 50 |
| Recommended value: | 2048 |
| Value exists by default? | **No**, needs to be added. |

**Note**

Start with the default or recommended values for these registry keys, and increase the value in small increments as needed. The more outstanding connections that exist, the more memory resources will be used by the server. If you set the values too high, the server could run out of resources such as paged pool memory.

For more information about setting these values, see Microsoft Knowledge Base article 317241 "How to troubleshoot Event ID 2021 and Event ID 2022" (http://go.microsoft.com/fwlink/?LinkID=105642) and Microsoft Knowledge Base article 810886 "The network BIOS command limit has been reached" error message in Windows Server 2003, in Windows XP, and in Windows 2000 Server (http://go.microsoft.com/fwlink/?LinkId=158215).

## Disable short-file-name (8.3) generation

When a long file name is created using the Windows NTFS file system, the default behavior is to generate a corresponding short file name in the older 8.3 DOS file name convention for compatibility with older operating systems. This functionality can be disabled through a registry entry, offering a performance increase.

**Caution**

Before disabling short name generation, ensure there are no DOS or 16-bit applications running on the server that require 8.3 file names.

### NTFSDisable8dot3NameCreation

| | |
|---|---|
| Key: | HKLM\SYSTEM\CurrentControlSet\Control\FileSystem |
| Value: | NTFSDisable8dot3NameCreation |
| Data Type: | REG_DWORD |
| Range: | 0 – 1 |
| Default value: | 0 |
| Recommended value: | 1 |
| Value exists by default? | **Yes** |

In Windows Server 2003, this value can be set by using the following command:

```
fsutil behavior set disable8dot3 1
```

### Optimize data throughput for network applications

If Windows Server is configured to optimize data throughput for network applications, the working set of BizTalk Server and other applications will have a priority over the working set of the file system cache. This setting is normally the best setting to use for all servers except dedicated file servers or with applications exhibiting file server-like characteristics.
To optimize data throughput for network applications follow these steps:

1. In Windows Explorer, right-click **My Network Places**, and then click **Properties**.
2. Right-click the Local Area Connection you want to optimize, and then click **Properties**.

3. In the **This connection uses the following items** list, click (but do not clear its check box) **File and Printer Sharing for Microsoft Networks**, and then click **Properties**.

4. Click **Maximize data throughput for network applications**, click **OK**, and then click **Close**.

Optimize data throughput for network applications can also be applied by setting the following registry entries to the recommended values:

### Size

| | |
|---|---|
| Key: | HKLM\System\CurrentControlSet\Services\LanmanServer\Parameters |
| Value: | Size |
| Data Type: | REG_DWORD |
| Recommended value: | 3 |
| Value exists by default? | **Yes** |

### LargeSystemCache

| | |
|---|---|
| Key: | HKLM\System\CurrentControlSet\Control\Session Manager\MemoryManagement |
| Value: | LargeSystemCache |
| Data Type: | REG_DWORD |
| Recommended value: | 0 |
| Value exists by default? | **Yes** |

### Disable random driver verification

The driver verifier at random intervals verifies drivers for debugging. Disabling this functionality might improve system performance. For many high-throughput systems, every CPU cycle counts. Disable random driver verification with the following registry entry:

### DontVerifyRandomDrivers

| | |
|---|---|
| Key: | HKLM\SYSTEM\CurrentControlSet\Control\FileSystem |
| Value: | DontVerifyRandomDrivers |
| Data Type: | REG_DWORD |
| Range: | 0 – 1 |
| Default value: | 0 |
| Recommended value: | 1 |
| Value exists by default? | **No**, needs to be added. |

### Disable NTFS last access updates

Each file and folder on an NTFS volume includes an attribute called Last Access Time. This attribute shows when the file or folder was last accessed, such as when a user performs a folder listing, adds files to a folder, reads a file, or makes changes to a file. Maintaining this information creates performance overhead for the file system especially in environments where a large number of files and directories are accessed quickly and in a short period of time, for example when using the BizTalk File Adapter. Apart from in highly secure environments, retaining this information might add a burden to a server that can be avoided by updating the following registry key:

### NTFSDisableLastAccessUpdate

| | |
|---|---|
| Key: | HKLM\SYSTEM\CurrentControlSet\Control\FileSystem |
| Value: | NTFSDisableLastAccessUpdate |
| | |

| | |
|---|---|
| Data Type: | REG_DWORD |
| Range: | 0 – 1 |
| Default value: | 0 |
| Recommended value: | 1 |
| Value exists by default? | **No**, needs to be added. |

In Windows Server 2003 and Windows Server 2008, this value can be set by using the following command:

```
fsutil behavior set disablelastaccess 1
```

For more information about disabling NTFS last access update on Windows Server 2003, see the Windows Server 2003 Deployment Guide (http://go.microsoft.com/fwlink/?LinkId=158216).

For more information about disabling NTFS last access update on Windows Server 2008 see Fsutil behavior (http://go.microsoft.com/fwlink/?LinkId=160067) on MSDN.

## Applying registry settings with the operating system optimization Windows PowerShell script

This guide includes a Windows PowerShell script that can be executed on each computer in the BizTalk Server environment to apply registry settings that will optimize operating system performance, using the recommended values discussed in this topic. To run this script follow these steps:

1. **Install Windows PowerShell** – Windows PowerShell can be downloaded from How to Download Windows PowerShell 1.0 (http://go.microsoft.com/fwlink/?LinkId=77521). PowerShell must be installed in order to run the operating system optimization script.

2. **Run the operating system optimization script**

    a. Copy the script from the "Optimizing operating system performance registry settings" section of Windows PowerShell Scripts into notepad and save as Set-OSRegSettings.ps1.

    b. Launch PowerShell and change directories to the folder that contains the saved script.

    c. Execute the script with the following command:

```
.\Set-OSRegSettings.ps1 –RAMMb <Installed memory in MB> -NumCPU <number o
```

> **Note**
>
> If the script does not run, or opens in Notepad instead of running, ensure the PowerShell execution policy permits running PowerShell scripts. To determine the current PowerShell execution policy run the **Get-ExecutionPolicy** PowerShell command. To change the current PowerShell execution policy run the **Set-ExecutionPolicy** PowerShell command.

The operating system optimizations PowerShell script generates a log file named "OSSettings.log" in the directory from which the script was executed. This log file details which values were changed and lists the original value as well as the new value. For simplicity sake, and so that all logs are accessible from the same place, it is recommended that this script is placed in a networked file share and that the script is run from that file share on all computers in the BizTalk Server environment.

# See Also

### Concepts

Windows PowerShell Scripts