

Microsoft Dynamics® AX

# Planning database configuration for Microsoft Dynamics AX

White Paper

This document provides a checklist for Windows Server and SQL Server configuration for optimal performance of the Microsoft Dynamics AX 2009 application.

Date: March 10, 2010

<http://www.microsoft.com/dynamics/ax/>



# Table of Contents

<b>Introduction.....</b>	<b>3</b>
Audience.....	3
Prerequisites .....	3
<b>Planning database storage.....</b>	<b>3</b>
<b>Planning SQL Server infrastructure.....</b>	<b>4</b>
Assumptions .....	4
<b>Checklist for Windows Server .....</b>	<b>4</b>
<b>Service configuration of the SQL Server instance.....</b>	<b>5</b>
<b>Configuration of the SQL Server instance.....</b>	<b>5</b>
Memory architecture .....	5
The <i>max degree of parallelism</i> option .....	6
The <i>max server memory</i> option .....	6
Memory for large databases in a 32-bit environment .....	7
Monitor available memory .....	7
Tempdb database storage configuration.....	7
Microsoft Dynamics AX database configuration .....	8
Physical Storage Configuration .....	9

## Introduction

Appropriate planning of database infrastructure is a critical requirement for optimal performance when using Microsoft Dynamics AX. This topic provides a high-level view of factors you need to consider when planning storage for your Microsoft Dynamics AX database. You will also need to follow carefully the design guidelines of your chosen database server when planning storage.

This topic provides details based on Microsoft SQL Server best practices for online transaction processing (OLTP) databases. For detailed information about Oracle Database Server, consult the Oracle documentation. For your business intelligence and reporting requirements, refer to the SQL Server documentation for SQL Server Reporting Services and SQL Server Analysis Services.

The Infrastructure Planning and Design (IPD) series provides guidance for Microsoft infrastructure products. The series is a collection of documents that leads the reader through a sequence of core decision points to design infrastructure for Microsoft products. The SQL Server 2008 guide is available on [TechNet](#).

## Audience

This white paper is designed for Microsoft Dynamics AX administrators and SQL Server database administrators who are responsible for administration of the Microsoft Dynamics AX application.

## Prerequisites

To benefit from this white paper, you should have knowledge in the following areas:

- Windows Server administration
- SQL Server administration including advanced configuration options, memory management, performance management, and troubleshooting
- Microsoft Dynamics AX system administration

## Planning database storage

Designing a data storage solution involves multiple and integrated aspects. We recommend that you follow these ordered steps when you approach this task.

1. Characterize the input/output (I/O) load of the Microsoft Dynamics AX application. The I/O characteristics will depend on your business requirements and the Microsoft Dynamics AX modules and components you plan to deploy. Some of the questions you can ask to determine your I/O characteristics are:
  - a) What is the read versus write ratio of the application?
  - b) What is the typical I/O volume (I/O per second)?
  - c) How much of the I/O is sequential in nature and how much is random in nature?
2. Determine availability and performance requirements for the database system.
3. Determine the hardware required to support the decisions made in steps 1 and 2.
4. Configure SQL Server to take advantage of the hardware in step 3.
5. Track performance as workload changes.

Step-by-step guidance on database architecture and storage is beyond the scope of this document. See Microsoft SQL Server [Storage Top 10 Best Practices](#) and [Physical Database Storage Design](#) for more information.

## Planning SQL Server infrastructure

Configuration of Windows Server and SQL Server greatly impacts the performance of the Microsoft Dynamics AX database. This section provides detailed configuration checklists for Windows Server and SQL Server.

### Assumptions

The configuration checklist recommendations in this document assume:

- That you are following the documented best practices for Windows Server and SQL Server.
- You are using a dedicated SQL Server 2005 or SQL Server 2008 server.
- You are using a single instance of SQL Server that is dedicated to running the Microsoft Dynamics AX production database. We recommend that you store your test and development databases on a separate server from the production database.

The next sections in this document provide detailed checklists for Windows Server and SQL Server configuration settings.

### Checklist for Windows Server

In addition to documented best practices for Windows Server, consider the following checklist. For instructions, refer to Windows Server documentation.

- Configure Windows Server to provide the best performance for background services instead of interactive applications (programs). SQL Server runs as a background service, so this setting is optimal for SQL Server.

Use the following steps to configure Windows Server 2008:

- A. In Windows Server 2008, from the **Control Panel**, double-click **System**. In Windows Server 2008 R2, click **Control Panel > System and Security > System**.
- B. Click **Advanced system settings**.
- C. On the **Advanced** tab, under **Performance**, click **Settings**.
- D. On the **Advanced** tab, under **Processor scheduling**, select **Background services**. Click **OK**.

For Windows Server 2003 instructions, see [Change processor scheduling options](#).

- In Windows Server 2003, set the **Specify memory usage** option to **Programs**. This setting is optimal for SQL Server because SQL Server maintains its own cache. The memory usage setting **System cache** is more suitable for file and print servers and does not apply to SQL Server. For instructions, see [Specify memory usage](#).

**Note:** This setting is not required for Windows Server 2008.

- If you are using Windows Server 2003 with AMD processors, ensure that boot.ini contains the parameter /USEPMTIMER. For more information, see [Explanation for the USEPMTIMER switch in the boot.ini](#) and the knowledge base article [895980](#).

**Note:** This setting is not required for Windows Server 2008.

## Service configuration of the SQL Server instance

In addition to the standard best practices for SQL Server, consider the following configuration settings for the SQL Server service:

- Run the SQL Server service under a Windows Active Directory domain account with the minimum necessary privileges. For more information, see [SQL Server 2005 Security Overview for Database Administrators](#) and [SQL Server 2008 Security Overview for Database Administrators](#).
- Grant the SQL Server service account the **Lock pages in memory** privilege. This privilege is not automatically granted to local administrators. For instructions, see [How to: Enable the Lock Pages in Memory Option \(Windows\)](#). For more information, see:
  - The Microsoft Customer Service and Support (CSS) SQL Server Engineers [blog](#)
  - On the Microsoft Support site, [How to reduce paging of buffer pool memory in the 64-bit version of SQL Server](#)
  - [Slava Oks's WebLog](#)
- Instant file initialization is only available if the SQL Server (MSSQLSERVER) service account has been granted the SE\_MANAGE\_VOLUME\_NAME right. Members of the Windows Administrator group have this right and can grant it to other users by adding them to the **Perform Volume Maintenance Tasks** security policy. For more information about assigning user rights, see the Windows Server documentation. For more information, see [Database file initialization](#).
- Enable the TCP/IP network protocol. Depending upon the SQL Server edition selected, this protocol may be automatically installed during the installation of SQL Server. For instructions, refer to SQL Server documentation. For SQL Server 2005 instructions, see [Choosing a Network Protocol](#). For SQL Server 2008 instructions, see [How to: Enable or Disable a Server Network Protocol \(SQL Server Configuration Manager\)](#).
- Disable hyperthreading. Microsoft Dynamics AX has not been extensively tested with hyperthreading and we recommend that you disable it. For instructions, refer to the hardware documentation for your server as this step must be performed in the BIOS settings of the server.

## Configuration of the SQL Server instance

In addition to documented best practices for SQL Server, consider the following configuration settings for the SQL Server instance.

### Memory architecture

SQL Server supports Address Windowing Extensions (AWE), allowing use of more than four gigabytes of physical memory on 32-bit versions of Microsoft Windows operating systems. By using AWE and the **Locked Pages in Memory** privilege, you can specify the amount of memory provided to the SQL Server Database Engine. Also, the AWE mechanism and locked pages may enable better performance in a 64-bit SQL Server environment. For more information, see [Memory Architecture](#).

## The *max degree of parallelism* option

The **max degree of parallelism** configuration is an instance-wide setting. Microsoft Dynamics AX OLTP workloads generally perform better when intra-query parallelism is disabled. However, the upgrade process benefits from parallelism. Use the following settings to configure the value for a normal operation or an upgrade:

- During normal operation of the Microsoft Dynamics AX database, set the **max degree of parallelism** to 1.
- During an upgrade to a new release of Microsoft Dynamics AX, set the **max degree of parallelism** to the least of:
  - Eight
  - The number of physical processor cores
  - The number of physical processor cores per non-uniform memory access (NUMA) node
- Use the following query to set the **max degree of parallelism** value. Examine the output from the second `sp_configure` statement and confirm that the new value is the same as the run-value column for the **max degree of parallelism** setting. For example, the first `sp_configure` statement in the following code will set the value to 1. The second `sp_configure` statement will return a value of 1 which is same as the run-value of the first `sp_configure` statement.

```
EXEC sp_configure 'show advanced options', 1;
RECONFIGURE;
GO
EXEC sp_configure 'max degree of parallelism', 1;
RECONFIGURE;
GO
EXEC sp_configure;
```

For more information for SQL Server 2008, see [max degree of parallelism Option](#). For more information for SQL Server 2005, see [max degree of parallelism Option](#). For general guidelines, see [General guidelines to use to configure the MAXDOP option](#). For tips from the SQL Server team, visit the SQL Server Relational Engine team's blog, [SQL Server Engine Tips](#).

## The *max server memory* option

SQL Server dynamically acquires and frees memory as required. Typically, an administrator does not have to specify how much memory should be allocated to SQL Server, yet the option still exists and is required in some environments. Be sure that sufficient memory is available for the operation of Windows Server. For more information, see the section [Monitor available memory](#).

If you find that the dynamic allocation of memory adversely affects the operation of Windows Server, adjust the **max server memory** according to available random access memory (RAM). For more information, see [Effects of min and max server memory](#).

## Memory for large databases in a 32-bit environment

Consider using Address Windowing Extensions (AWE) for SQL Server in a 32-bit Microsoft Windows operating system environment. AWE addresses a limitation inherent in 32-bit applications (a 32-bit pointer cannot hold a memory address larger than four GB.). Using AWE, applications can directly reserve physical memory up to the maximum amount allowed by the operating system as non-paged memory. Using AWE allows SQL Server to cache more information instead of reading it from system paging files on disk. This caching yields performance gains through faster data access and reduced the frequency of disk access. For more information, see [Managing Memory for Large Databases](#).

The AWE mechanism and locked pages may enable better performance in a 64-bit SQL Server environment. For more information, see [Memory Architecture](#).

## Monitor available memory

Be sure that sufficient memory is available for the operation of the Windows Server. For example, be sure to run a dedicated instance of SQL Server on a server with at least four gigabytes of memory. If the available memory for the server goes below 500 megabytes (MB) for sustained periods, the server may experience performance degradation. Use the Windows Server operating system performance counter **Memory: Available Mbytes** to determine if the available memory drops below 500 MB for sustained periods. If the available memory drops below 500 MB frequently or for a sustained period of time, consider reducing the SQL Server **max server memory** setting or increasing the physical memory in the server.

Detailed guidance on memory management is beyond the scope of this document. For more information on monitoring memory and troubleshooting performance issues, refer to the Windows Server and SQL Server documentation.

## Tempdb database storage configuration

Determine total size of data and transaction log required for tempdb. Be sure to avoid the automatic growth (autogrow) operation. Determine the number of data files required using the following steps:

- Determine the number of processors that are available to SQL Server. Unless you are using an affinity mask, this number is same as the total number of processors you see in the **Performance** tab of the **Windows Task Manager**. When hyperthreading is not enabled, each processor corresponds to a processor core. Affinity masks and processor cores are beyond the scope of this document. For more information, see the Windows Server and SQL Server documentation.
- SQL Server creates one visible scheduler to process user requests for each processor. We recommend that you maintain one tempdb data file per processor based on performance testing of the Microsoft Dynamics AX online transaction processing (OLTP) workload. For more information, see the performance benchmark reports on [PartnerSource](#) or [CustomerSource](#). CustomerSource is a password-protected site for customers using Microsoft Dynamics and related business products. PartnerSource is a password-protected site for partners who focus on Microsoft Dynamics and related products. Both sites require you to log on.
- Isolating tempdb on dedicated storage can improve SQL Server performance. Consider moving the tempdb primary data file and log file to high-speed storage (if available). Microsoft Dynamics AX database runs in the read committed snapshot isolation (RCSI) mode. RCSI stores row versions in tempdb. Creating multiple files for tempdb data, even if these files reside on the same storage device, can improve performance of tempdb operations.
- Following SQL Server best practices, determine the size for the tempdb data and log files. For instructions, see [Working with tempdb in SQL Server 2005](#) or [Optimizing tempdb performance](#) (for SQL Server 2008). You must create one primary data file and one log file. Determine how many additional secondary data files you will need for the tempdb data. For best results, create data files of equal size. The total number of your data files must equal the total number of processor cores. The aggregate size of all data files, including the primary data file, must equal the total data size you determined for the tempdb database.

- Resize the tempdb primary data file and log file at this time. Move the primary data file and log file to dedicated storage, if available. The primary tempdb data file cannot be moved while the SQL Server instance is running. You must alter the database and restart SQL Server instance to complete the move. For instructions, consult the SQL Server documentation for the ALTER DATABASE transact-SQL statement.

**NOTE:** The tempdb data and transaction log files may reside on the same storage device. Create the additional data files for tempdb either from the SQL Server Management Studio or by using the ALTER DATABASE statement.

- If space is available on the drive where tempdb files are allocated, configure the autogrow property in megabytes instead of percentage for both data and log files. If possible, configure the data and log files to grow by 100 to 500 MB, depending on available space. Autogrow should not be used for space management, but as a safety mechanism to enable tempdb to grow in case tempdb files exhaust their initially allocated space. Monitor the data files, and when they grow, adjust the initial allocation to avoid automatic growth in the future. Configuring autogrow in MB rather than percent increments allow for more predictable space allocation and helps avoid extremely small or large growth increments.
- Monitor the tempdb data and log files to ensure that they are all sized correctly and that all data files are of equal size. Use SQL Server Management Studio or a transact-SQL query to view the database properties. Verify that all the data files are of equal size and have the same size as the initial value you provided. If one or more files have grown, adjust the initial size for all files.

## Microsoft Dynamics AX database configuration

Consider the following configuration settings for the Microsoft Dynamics AX database. Use the SQL Server Management Studio or an appropriate ALTER DATABASE statement to configure these settings. For instructions, refer to the SQL Server documentation. This document uses the SQL Server 2008 conventions for the following settings:

- Set COMPATIBILITY\_LEVEL to 90 for SQL Server 2005 or 100 for SQL Server 2008.
- Set READ\_COMMITTED\_SNAPSHOT to on. Performance testing of the Microsoft Dynamics AX application shows superior performance with the READ\_COMMITTED\_SNAPSHOT isolation option set to on. You cannot set this option through the SQL Server Management Studio and must use an ALTER DATABASE statement.
  - a) Use the following query, with no other active connections in the database, where <database name> is the name of the Microsoft Dynamics AX database:
 

```
ALTER DATABASE <database name>
    SET READ_COMMITTED_SNAPSHOT ON;
```
  - b) Query the sys.databases catalog view and verify that the Microsoft Dynamics AX database contains a value of 1 in the is\_read\_committed\_snapshot\_on column. For more information on the sys.databases catalog view, see the SQL Server documentation for [SQL Server 2008](#) and [SQL Server 2005](#). For more information on choosing row-versioning-based isolation levels, see SQL Server documentation for [SQL Server 2008](#) and [SQL Server 2005](#).
- Set AUTO\_CREATE\_STATISTICS and AUTO\_UPDATE\_STATISTICS to on. Set AUTO\_UPDATE\_STATISTICS\_ASYNC to off. Performance testing of the Microsoft Dynamics AX application demonstrates superior performance with these configuration settings.

- Ensure that the AUTO\_SHRINK option is set to off. Automatically shrinking database files will result in performance degradation. Consider a pre-defined schedule for the database administrator to manually shrink the database files. For more information, see the [SQL Server Storage Engine Team's blog](#).
- If the database is configured to grow automatically, use megabytes instead of the percentage value for both data and log files. Depending upon available space, use an increment value of 100 to 500 MB. Using a specific size in megabytes provides for a predictable and anticipated growth size. A percentage-based growth setting can result in either too many small increments in rapid succession or overly large increments.

The automatic growth option should be considered a safety mechanism that allows the database files to grow when absolutely necessary and thus prevent errors. The automatic growth option should not be used as a long-term storage management solution. The database administrator must monitor and respond to an automatic growth event by adjusting the size of all database files to minimize the probability of future automatic growth events. Create a routine maintenance schedule to proactively monitor and expand the data file sizes to prevent an automated automatic growth event.

## Physical Storage Configuration

This section provides general recommendations for physical storage. Determine the applicability of these recommendations in your environment. Some storage area network (SAN) vendors may have alternate recommendations that take precedence. Recommendations are listed by priority.

- Many factors contribute to optimal disk I/O performance. For disk partitions created with Windows Server 2003, properly correlating partition alignment with stripe unit size and file allocation unit size is a best practice and provides a foundation for optimal performance.

Windows Server 2008 aligns partitions by default. When you upgrade to Windows Server 2008, preexisting partitions are not automatically aligned and must be manually rebuilt for optimal performance. Therefore, until you rebuild the migrated partitions from Windows Server 2003, disk partition alignment will remain a relevant technology.

Check existing disks on the server and be aware of the differences in analysis of basic partitions and dynamic volumes. Rebuild the partitions where possible and appropriate and create all new partitions based on guidance from the SAN vendor. In the absence of vendor-specific recommendations, follow SQL Server best practices. For SQL Server 2008, see [Disk Partition Alignment Best Practices for SQL Server](#). For SQL Server 2005, see [Predeployment I/O Best Practices](#) and [Physical Database Storage Design](#).

The partition offset value must be a multiple of the stripe size; that is, the expression (partition offset / stripe size) must resolve to an integer value.

- Create the tempdb database files, Microsoft Dynamics AX database data files, and Microsoft Dynamics AX log files on disk arrays of type RAID 1, RAID 0 + 1, or RAID 10. We recommend RAID 10 for these files; do not use RAID 5. For more information, refer to Windows Server and SQL Server documentation.
- Store the Microsoft Dynamics AX database data files on separate physical stores from the transaction log files.
- Store the tempdb data files on a separate physical store from the data and log files of the Microsoft Dynamics AX database.
- Store other database files on separate physical stores from the data and log files of tempdb and the Microsoft Dynamics AX database.

---

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

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

Worldwide +1-701-281-6500

[www.microsoft.com/dynamics](http://www.microsoft.com/dynamics)

This document is provided "as-is." Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it.

Some examples depicted herein are provided for illustration only and are fictitious. No real association or connection is intended or should be inferred.

This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes. You may modify this document for your internal, reference purposes.

Microsoft, Active Directory, Microsoft Dynamics, Windows, Windows Server, SQL Server and the Microsoft Dynamics Logo are trademarks of the Microsoft group of companies. All other trademarks are property of their respective owners.

© 2009 Microsoft Corporation. All rights reserved.

**Microsoft**