



Microsoft Dynamics NAV 2013 SQL Readiness Training

Module 1: General Configuration Considerations

Microsoft®

Conditions and Terms of Use

Microsoft Confidential

This training package content is proprietary and confidential, and is intended only for users described in the training materials. This content and information is provided to you under a Non-Disclosure Agreement and cannot be distributed. Copying or disclosing all or any portion of the content and/or information included in this package is strictly prohibited.

The contents of this package are for informational and training purposes only and are provided "as is" without warranty of any kind, whether express or implied, including but not limited to the implied warranties of merchantability, fitness for a particular purpose, and non-infringement.

Training package content, including URL and other Internet Web site references, is subject to change without notice. Because Microsoft must respond to changing market conditions, the content should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication. Unless otherwise noted, the companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted herein are fictitious, and no association with any real company, organization, product, domain name, e-mail address, logo, person, place, or event is intended or should be inferred.

Copyright and Trademarks

© 2012 Microsoft Corporation. All rights reserved.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

For more information, see **Use of Microsoft Copyrighted Content** at <http://www.microsoft.com/about/legal/permissions/>

Microsoft®, Microsoft Dynamics®, SQL Server®, Windows Server®, Vista®, Internet Explorer®, and Windows® are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Microsoft products mentioned herein may be either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. All other trademarks are property of their respective owners.

About the Authors

Author:

Gerard Conroy

Bio:

Gerard is an Escalation Engineer in the Microsoft Dynamics NAV support team based in the United Kingdom. He has been working with Microsoft Dynamics NAV since 2007 and his previous roles within Microsoft include the SQL Server support team and the internal IT Department.

Author:

Paulo Condeça

Bio:

Paulo is a Microsoft Premier Field Engineer based in Portugal.

Author:

Pedro Lopes

Bio:

Pedro is a Premier Field Engineer based in Portugal.

Table of Contents

OVERVIEW	1
LESSON 1: EVALUATE THE SQL SERVER SOFTWARE REQUIREMENTS FOR MICROSOFT DYNAMICS NAV 2013:	2
LESSON 2: EVALUATE THE SQL SERVER HARDWARE AND OPERATING SYSTEM REQUIREMENTS FOR MICROSOFT DYNAMICS NAV 2013.....	3
LESSON 3: WINDOWS AND MACHINE SETTINGS	11
CPU	11
Network.....	12
Memory	13
Storage	13
LESSON 4: SQL SERVER SETTINGS.....	16
SQL Server Parameters.....	16
SQL Server Files	18
Database Recommended Settings	22
SQL Server Maintenance	25
LESSON 5: MICROSOFT DYNAMICS NAV SETTINGS.....	27
LESSON 6: THE MICROSOFT DYNAMICS NAV 2013 SECURITY MODEL	29

Overview

In this module, we will discuss a basic Installation for Microsoft Dynamics NAV 2013 running with SQL Server 2012.

What You Will Learn

After completing this module, you will be aware of the Installation options for Microsoft Dynamics NAV 2013, which are directly relevant to the SQL Server platform.

One of the most important steps in a Microsoft Dynamics NAV implementation is the selection of the hardware and software platform.

When planning a Microsoft Dynamics NAV 2013 installation, you must make sure that the computers used meet the minimum requirements and are sufficient for your current and future needs. Failure to meet these specifications can cause the installation of some or all of the components to fail.

This module describes the requirements for installing Microsoft Dynamics NAV 2013. Understanding the setup requirements and options help you plan a successful installation. We will focus on the following topics:

- Evaluate the SQL Server software requirements for Microsoft Dynamics NAV 2013.
- Evaluate the hardware requirements for Microsoft Dynamics NAV 2013.
- Evaluate the security model

Lesson 1: Evaluate the SQL Server Software Requirements for Microsoft Dynamics NAV 2013:

The database components for Microsoft Dynamics NAV 2013 run on the following versions and editions of SQL Server:

- Microsoft SQL Server 2012 Express, Workgroup, Standard, or Enterprise (64-bit editions only).
- Microsoft SQL Server 2008 R2 Express, Workgroup, Standard, or Enterprise (64-bit editions only).
- Microsoft SQL Server 2008 Express, Workgroup, Standard, or Enterprise (64-bit editions only).

When installing the Demo version of Microsoft Dynamics NAV, if SQL Server 2008 R2, SQL Server 2008, or SQL Server 2012 is present on the computer, and there is an existing SQL Server instance named NAVDEMO, then the CRONUS International Ltd. demonstration database is installed in this instance. Otherwise, Setup automatically installs SQL Server 2012 Express, then creates a NAVDEMO instance, then installs CRONUS International Ltd. demonstration database in this instance.

Lesson 2: Evaluate the SQL Server Hardware and Operating System Requirements for Microsoft Dynamics NAV 2013

The Hardware and Operating system requirements for SQL Server are documented in detail in the SQL Server Books Online, which is the best source of information for anything related to the SQL Server product.

There are no SQL Server hardware requirements specific to Microsoft Dynamics NAV 2013 so refer to SQL Books Online for details. The following shows the requirements from Books Online for the SQL Server 2008 R2 Editions supported with Microsoft Dynamics NAV 2013.

SQL Server 2008 R2 Enterprise (64-bit) IA64

The following table shows system requirements for SQL Server 2008 R2 Enterprise (64-bit) IA64.



Note:

This edition of SQL Server 2008 R2 is not available in the following localized versions: Italian, Spanish, Brazilian (Portuguese), or Russian.

Component	Requirement
Processor	Processor type: <ul style="list-style-type: none"> Itanium processor or faster Processor speed: <ul style="list-style-type: none"> Recommended: 1.0 GHz or faster
Operating system	Windows Server 2008 R2 64-bit Itanium Windows Server 2008 SP2 64-bit Itanium Windows Server 2003 SP2 64-bit Itanium Datacenter Windows Server 2003 SP2 64-bit Itanium Enterprise
Memory	RAM: <ul style="list-style-type: none"> Minimum: 1 GB Recommended: 4 GB or more Maximum: Operating system maximum

SQL Server 2008 R2 Enterprise (64-bit) x64

The following table shows system requirements for SQL Server 2008 R2 Enterprise (64-bit) x64:

Component	Requirement
Processor	Processor type: <ul style="list-style-type: none"> • Minimum: AMD Opteron, AMD Athlon 64, Intel Xeon with Intel EM64T support, Intel Pentium IV with EM64T support Processor speed: <ul style="list-style-type: none"> • Minimum: 1.4 GHz • Recommended: 2.0 GHz or faster
Operating system	Windows Server 2003 SP2 64-bit x64 Datacenter Windows Server 2003 SP2 64-bit x64 Enterprise Windows Server 2003 SP2 64-bit x64 Standard Windows Server 2003 SP2 64-bit x64 Web Windows Server 2008 SP2 64-bit x64 Datacenter Windows Server 2008 SP2 64-bit x64 Datacenter without Hyper-V Windows Server 2008 SP2 64-bit x64 Enterprise, Windows Server 2008 SP2 64-bit x64 Enterprise without Hyper-V Windows Server 2008 SP2 64-bit x64 Standard Windows Server 2008 SP2 64-bit x64 Standard without Hyper-V Windows Server 2008 SP2 64-bit x64 Web Windows 2008 R2 64-bit x64 Datacenter Windows 2008 R2 64-bit x64 Enterprise Windows 2008 R2 64-bit x64 Standard Windows 2008 R2 64-bit x64 Web Windows Server 2008 R2 x64 for Windows Essential Server Solutions
Memory	RAM: <ul style="list-style-type: none"> • Minimum: 1 GB • Recommended: 4 GB or more • Maximum: Operating system maximum

SQL Server 2008 R2 Standard (64-bit) x64

The following table shows system requirements for SQL Server 2008 R2 Standard (64-bit) x64:

Component	Requirement
Processor	<p data-bbox="483 415 662 447">Processor type:</p> <ul data-bbox="532 468 1382 531" style="list-style-type: none"><li data-bbox="532 468 1382 531">• Minimum: AMD Opteron, AMD Athlon 64, Intel Xeon with Intel EM64T support, Intel Pentium IV with EM64T support <p data-bbox="483 552 678 583">Processor speed:</p> <ul data-bbox="532 604 959 688" style="list-style-type: none"><li data-bbox="532 604 792 636">• Minimum: 1.4 GHz<li data-bbox="532 657 959 688">• Recommended: 2.0 GHz or faster

Component	Requirement
Operating system ²	<p>Windows XP Professional SP2 x64</p> <p>Windows Server 2003 SP2 64-bit x64 Datacenter</p> <p>Windows Server 2003 SP2 64-bit x64 Enterprise</p> <p>Windows Server 2003 SP2 64-bit x64 Standard</p> <p>Windows Vista SP2 Ultimate x64</p> <p>Windows Vista SP2 Enterprise x64</p> <p>Windows Vista SP2 Business x64</p> <p>Windows Server 2008 SP2 x64 Datacenter, Windows Server 2008 SP2 x64 Datacenter without Hyper-V</p> <p>Windows Server 2008 SP2 x64 Enterprise, Windows Server 2008 SP2 x64 Enterprise without Hyper-V</p> <p>Windows Server 2008 SP2 x64 Standard, Windows Server 2008 SP2 x64 Standard without Hyper-V</p> <p>Windows Server 2008 SP2 x64 Web</p> <p>Windows Server 2008 SP2 for Windows Essential Server Solutions²</p> <p>Windows Server 2008 SP2 x64 Foundation Server</p> <p>Windows 7 x64 Ultimate</p> <p>Windows 7 x64 Enterprise</p> <p>Windows 7 x64 Professional</p> <p>Windows Server 2008 R2 64-bit x64 Datacenter</p> <p>Windows Server 2008 R2 64-bit x64 Enterprise</p> <p>Windows Server 2008 R2 64-bit x64 Standard</p> <p>Windows Server 2008 R2 64-bit x64 Web</p> <p>Windows Server 2008 R2 x64 for Windows Essential Server Solutions</p> <p>Windows Server 2008 R2 64-bit x64 Foundation Server</p>
Memory	<p>RAM:</p> <ul style="list-style-type: none"> • Minimum: 1 GB • Recommended: 4 GB or more • Maximum: 64 GB

SQL Server 2008 R2 Express x64 (64-bit)

The following table shows system requirements for SQL Server 2008 R2 Express x64 (64-bit):

Component	Requirement
Processor	<p data-bbox="479 268 657 300">Processor type:</p> <ul data-bbox="527 321 1377 384" style="list-style-type: none"><li data-bbox="527 321 1377 384">• Minimum: AMD Opteron, AMD Athlon 64, Intel Xeon with Intel EM64T support, Intel Pentium IV with EM64T support <p data-bbox="479 405 673 436">Processor speed:</p> <ul data-bbox="527 457 950 541" style="list-style-type: none"><li data-bbox="527 457 787 489">• Minimum: 1.4 GHz<li data-bbox="527 510 950 541">• Recommended: 2.0 GHz or faster

Component	Requirement
Operating system	<p>Windows Server 2003 x64</p> <p>Windows Server 2003 SP2 64-bit x64 Datacenter</p> <p>Windows Server 2003 SP2 64-bit x64 Enterprise</p> <p>Windows Server 2003 SP2 64-bit x64 Standard</p> <p>Windows Vista SP2 Ultimate x64</p> <p>Windows Vista SP2 Home Premium x64</p> <p>Windows Vista SP2 Home Basic x64</p> <p>Windows Vista SP2 Enterprise x64</p> <p>Windows Vista SP2 Business x64</p> <p>Windows Server 2008 SP2 64-bit x64 Datacenter</p> <p>Windows Server 2008 SP2 64-bit x64 Datacenter without Hyper-V</p> <p>Windows Server 2008 SP2 64-bit x64 Enterprise</p> <p>Windows Server 2008 SP2 64-bit x64 Enterprise without Hyper-V</p> <p>Windows Server 2008 SP2 64-bit x64 Standard</p> <p>Windows Server 2008 SP2 64-bit x64 Standard without Hyper-V</p> <p>Windows Server 2008 SP2 64-bit x64 Web</p> <p>Windows Server 2008 SP2 x64 Foundation Server</p> <p>Windows 7 x64 Ultimate</p> <p>Windows 7 x64 Home Premium</p> <p>Windows 7 x64 Home Basic</p> <p>Windows 7 x64 Enterprise</p> <p>Windows 7 x64 Datacenter</p> <p>Windows Server 2008 R2 64-bit x64 Datacenter</p> <p>Windows Server 2008 R2 64-bit x64 Enterprise</p> <p>Windows Server 2008 R2 64-bit x64 Standard</p> <p>Windows Server 2008 R2 64-bit x64 Web</p> <p>Windows Server 2008 R2 x64 for Windows Essential Server Solutions</p> <p>Windows Server 2008 R2 64-bit x64 Foundation Server</p>
Memory	<p>RAM:</p> <ul style="list-style-type: none"> • Minimum:256 MB • Recommended: 1.024 GB • Maximum: 1 GB for the Database Engine

Cross-Language Support



More:

For more information about cross-language support and considerations for installing SQL Server in localized languages, see:

Local Language Versions in SQL Server

Hard Disk Space Requirements

During installation of SQL Server 2008 R2, Windows Installer creates temporary files on the system drive. Before you run Setup to install or upgrade SQL Server, verify that you have at least 2.0 GB of available disk space on the system drive for these files. This requirement applies even if you install SQL Server components to a non-default drive.

Actual hard disk space requirements depend on your system configuration and the features that you decide to install. The following table provides disk space requirements for SQL Server 2008 R2 components:

Feature	Disk space requirement
Database Engine and data files, Replication, and Full-Text Search	711 MB
Analysis Services and data files	345 MB
Reporting Services and Report Manager	304 MB
Integration Services	591 MB
Client Components (Other than Books Online and Integration Services tools)	1823 MB
SQL Server Books Online	157 MB

Installing SQL Server on a Domain Controller

For security reasons, Microsoft recommends that you do not install SQL Server 2008 R2 on a domain controller. SQL Server Setup will not block installation on a computer that is a domain controller, but the following limitations apply:

- On Windows Server 2003, SQL Server services can run under a domain account or a local system account.
- You cannot run SQL Server services on a domain controller under a local service account or a network service account.
- After SQL Server is installed on a computer, you cannot change the computer from a domain member to a domain controller. You must uninstall SQL Server before you change the host computer to a domain controller.

- After SQL Server is installed on a computer, you cannot change the computer from a domain controller to a domain member. You must uninstall SQL Server before you change the host computer to a domain member.
- SQL Server failover cluster instances are not supported where cluster nodes are domain controllers.
- SQL Server Setup cannot create security groups or provision SQL Server service accounts on a read-only domain controller. In this scenario, Setup will fail.

Supported Clients (64-Bit)

SQL Server 2008 R2 64-bit client components can be installed on Windows Server 2003 (64-bit).

Lesson 3: Windows and Machine Settings

The following are some guidelines for Windows and Machine settings provided by Microsoft Premier Field Engineering. These are optimization suggestions for high performance installations and not hard and fast requirements for a successful Microsoft Dynamics NAV 2013 deployment. Consult the customer's system administrators about the most appropriate settings for their environment.

CPU

On Windows Server® 2008 or above, set the “High Performance” power plan in Control Panel -> Power Options -> OK. By default, Windows Server 2008 R2 sets the Balanced power plan, which enables energy conservation by scaling the processor performance based on current CPU utilization. On Intel X5500 and other last-generation CPUs, the clock is throttled down to save power (Processor P-state), and only increases when CPU utilization reaches a certain point. The Minimum and Maximum Processor Performance State parameters are expressed as a percentage of maximum processor frequency, with a value in the range 0 – 100. If a server requires ultra-low latency, invariant CPU frequency, or the very highest performance levels, such as a database server, it might not be helpful that the processors keep switching to lower-performance states. As such, the High Performance power plan caps the minimum processor performance state at 100 percent.

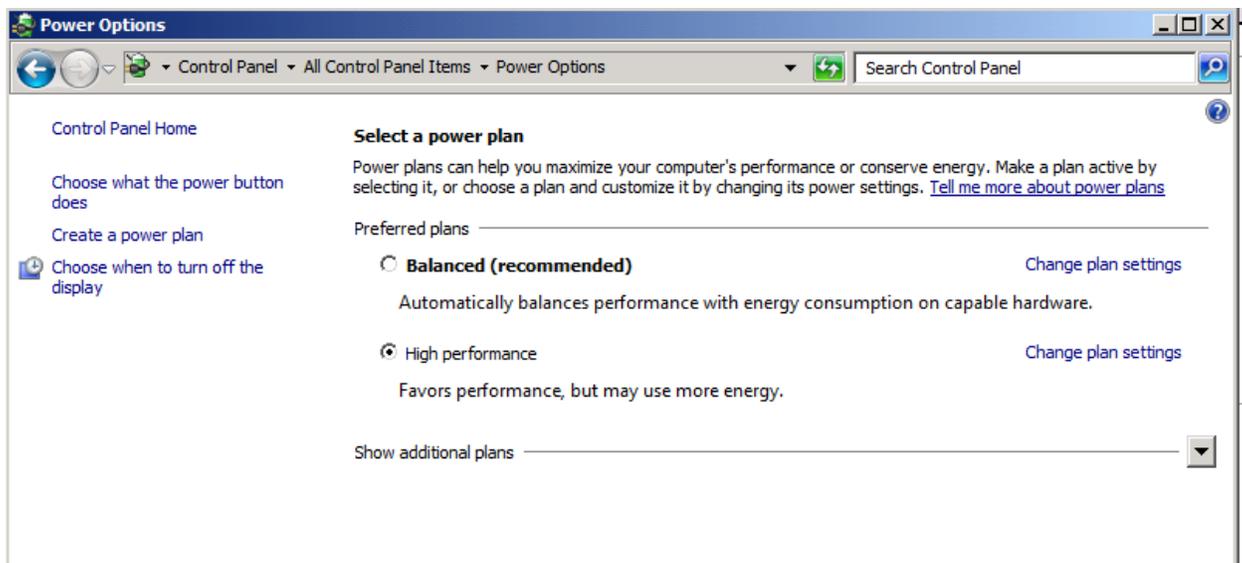


Figure 1: Control Panel showing the Power Options

The following are the built-in power plans and their common scenarios:

Plan	Description	Common applicable scenarios	Implementation highlights
Balanced	Default setting. Highest energy efficiency with minimum performance impact.	General computing.	Matches capacity to demand. Energy-saving features balance power and performance.
High Performance	Increases performance at the cost of high energy consumption. Should not be used unless absolutely necessary.	Low latency. Application code sensitive to processor frequency changes.	Processors are always locked at the highest performance state.

References and further reading material:

- [Degraded overall performance on Windows Server 2008 R2](#)
- [Windows Power Plans and CPU Performance](#)
- [Are your CPUs running slowly?](#)
- [Interpreting CPU Utilization for Performance Analysis](#)
- [SQL Server on Power-Saving CPUs? Not So Fast.](#)

Network

Configure NIC bindings in a MSFC, by setting the domain NICs at the top of the stack, followed by any private NIC (such as Heartbeat, if any). Domain needs to be accessible as the top network in the binding order or else performance problems might occur, as the network stack tries to access the domain and times out because it has to fail on private networks until it gets to the domain network in the binding list. If none of the network adapters has a domain name binding, then identify a domain NIC adapter from *Network Connections* and under *Properties* -> *Internet Protocol (TCP/IP)* -> *Properties* -> *Advanced* -> *DNS*, and add a DNS suffix for the domain adapter. After this, the domain NIC will have a domain name binding, so move this NIC adapter to the top of the network binding order.

References and further reading material:

- [Recommended private "Heartbeat" configuration on a cluster server](#)
- [Network adapter teaming and server clustering](#)
- [Network Binding Order Rule Warning in SQL Server 2008 Cluster Setup Explained](#)

- [SQL Server 2008 Failover Clustering White Paper](#)

Memory

Set the “Lock Pages in Memory” permission to the SQL Server account in Windows Server 2003 running on an x64 system. On SQL Server 2005 and above over Windows Server 2008 this assumes a much lower importance due to optimized memory management, and thus should only be used when necessary, namely if there are signs that *sqlservr* process is being paged out. This permission will allow for a copy of the buffer cache to stay resident in physical memory, preventing the system from paging the data to virtual memory on disk.

Also, setting this user right implies a restart of SQL Server service. Note that for 64-bit editions of SQL Server only Enterprise Edition can use the “Lock pages in memory” user right. This is applicable for SQL Server 2008 (RTM and SP1).

SQL Server 2008 SP1 CU2 introduce support for SQL Server Standard editions to use the “Lock pages in memory” user right, that also add the trace flag 845, that must be set up as a startup parameter to support page locking.

References and further reading material:

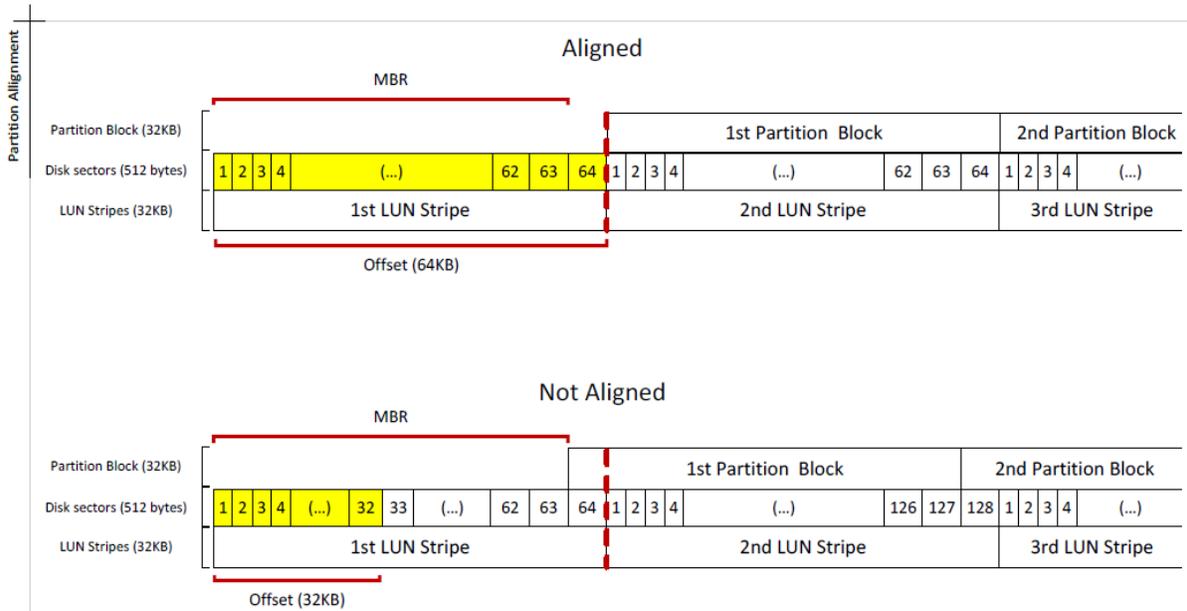
- [How to reduce paging of buffer pool memory in the 64-bit version of SQL Server](#)
- [Enabling Memory Support for Over 4 GB of Physical Memory](#)
- [How to: Enable the Lock Pages in Memory Option \(Windows\)](#)
- [Fun with Locked Pages, AWE, Task Manager, and the Working Set...](#)

Storage

Align LUN partition with the underlying disk clusters. Disk performance may be slower than expected when you use disks in Windows Server 2003. This issue may occur if the starting location of a partition is not aligned with the Stripe Unit size and Allocation Unit size (or Cluster size) in the disk partition that is created on RAID. A **64-KB offset (128 sectors) is a common value that works on many storage arrays** and has been used with success in the SQL Server test labs (at the time of writing this paper).

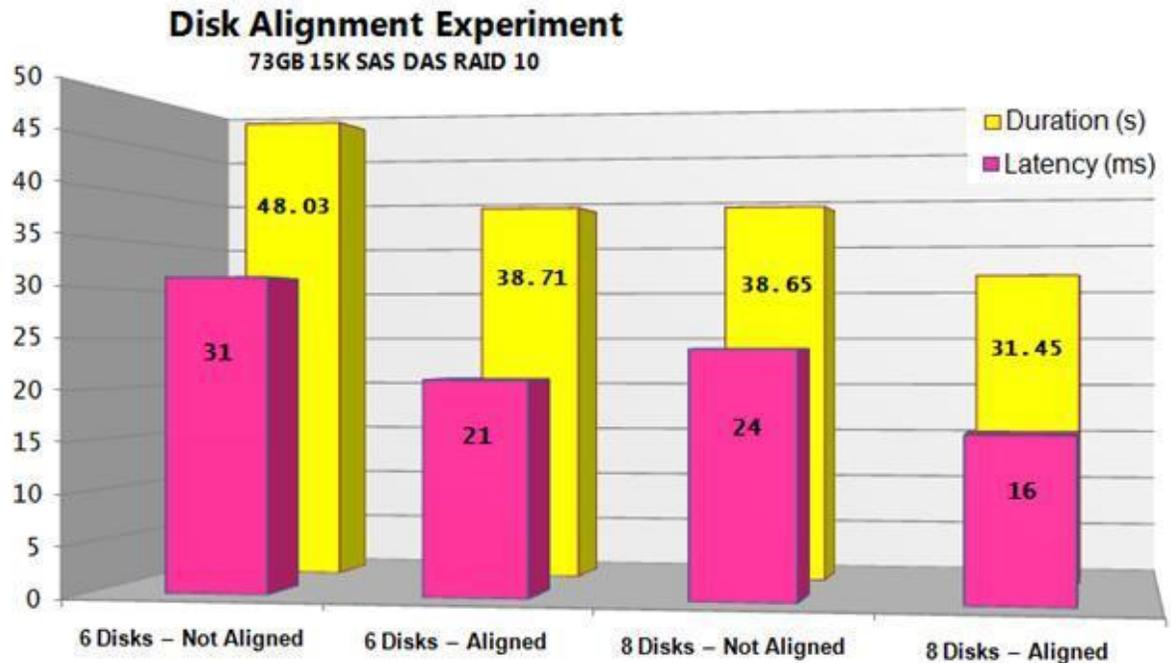
Partitions may contain hidden sectors and the total length of these may not coincide with a full track. For example, there are 63 hidden sectors in the MBR (Master Boot Record) so the first block of user data is written across the first and second tracks starting in the 64th sector, misaligning the entire partition. Subsequently, instead of one IOP (I/O operation) to read or write data, two IOPs are required because if the disk is not aligned, every Nth read

or write crosses a boundary, and the physical disk must perform two operations. Hence, the 64-KB offset alignment value. **Windows Server 2008 remove the need to manually establish correct alignment and use 1024-KB (2048 sectors) as a starting offset**, which will probably work well for almost any array, although this must be subject to the storage vendor approval. Note that Windows Server 2008 and Windows Server 2008 R2 the offset still defaults to 64-KB if the disk is smaller than 4 GB.



A brand new Windows Server 2008 install is not affected, but if the OS is upgraded in-place from Windows Server 2003 to Windows Server 2008, you will still experience this problem, because reconfiguring the hard disk offset setting requires the partitions to be re-created and reformatted. Therefore, this should be completed before any data is loaded.

Work with your hardware vendor to optimally configure the hard drives that support RAID devices.



References and further reading material:

- [Predeployment I/O Best Practices](#)
- [A Description of the Diskpart Command-Line Utility](#)
- [How to Use Diskpar.exe](#)
- [Disk Partition Alignment Best Practices for SQL Server](#)
- [Recommendations and Guidelines on configuring disk partitions for SQL Server](#)

Format the DATA and LOG partitions with NTFS 64-KB block size. SQL Server data pages are 8192 bytes, so the default NTFS block size (4096 bytes) reads only half a page and effectively doubles the number of I/O operations. Also, consider formatting the DATA drives in 64-KB blocks, because SQL Server commonly does an eight page read-ahead to improve performance during range and table scans.

References and further reading material:

- [Disk Partition Alignment Best Practices for SQL Server](#)

Lesson 4: SQL Server Settings

The following are some guidelines for SQL Server settings provided by Microsoft Premier Field Engineering. These are optimization suggestions for high performance installations and not hard and fast requirements for a successful Microsoft Dynamics NAV 2013 deployment. Consult the customer's database administrator about the most appropriate settings for their environment.

SQL Server Parameters

MAXDOP (“max degree of parallelism”). Most of the queries generated by Microsoft Dynamics NAV 2013 are simple TSQL statements which do not require parallel execution, e.g. very simple SELECT, UPDATE, INSERT, and DELETE SQL statements. This does not stop the compiler from sometimes attempting to use parallel execution when planning queries. This can sometimes negatively impact performance. However, with the introduction of the Query object in Microsoft Dynamics NAV 2013 some SQL Server queries will join several tables in a single SQL SELECT statement and therefore the possibility for performance benefits from parallel query execution exists. Our recommendation is to set MAXDOP to 1 initially to disable parallelism. When the system has been running for some time it may be beneficial to test the impact of parallelism on performance. Use the following query to identify the system generated recommendation for MAXDOP:

```
select case
    when cpu_count / hyperthread_ratio > 8 then 8
    else cpu_count / hyperthread_ratio
    end as optimal_maxdop_setting
from sys.dm_os_sys_info;
```

If any performance problems develop after enabling parallelism set MAXDOP back to 1 while troubleshooting.

References and further reading material:

- [Affinity mask Option](#)
- <http://blogs.msdn.com/b/nav/archive/2010/09/28/microsoft-dynamics-nav-sql-server-configuration-recommendations.aspx>

Limit the memory available for SQL Server by setting its “max_server_memory” server setting. Do not leave the default and ensure that the OS gets at least 1 GB, as recommended previously. As an example, these settings are for x64, on a dedicated database server, only running the DB engine:

Physical RAM	MaxServerMem Setting
2GB	1500
4GB	3200
6GB	4800
8GB	6400
12GB	10000
16GB	13500
24GB	21500
32GB	29000
48GB	44000
64GB	60000
72GB	68000
96GB	92000
128GB	124000

In the case of NUMA configurations, **if SQL Server is configured to run on a subset of the available NUMA nodes, use the max server memory setting to limit the buffer pool** to the memory available in the assigned NUMA nodes. For example, Dual-core or Quad-core CPUs will usually be grouped and acting as NUMA nodes by themselves, and that information is exposed by querying the available schedulers (sys.dm_os_schedulers).

At startup, the buffer pool attempts to get as much local memory for each node as possible, though it will frequently allocate memory from other NUMA nodes (foreign memory), since current Windows APIs cannot allocate memory from a specific node. Still, these pages are not initially used because they can frequently be transferred to the owning node and become local to that node.

When the value of max_server_memory is reached, some nodes may have foreign memory, but once the memory target is achieved, the buffer pool will treat local and foreign memory identically, even under memory pressure.

References and further reading material:

- [SQL Server® 2008 R2 Books Online: Growing and Shrinking the Buffer Pool Under NUMA](#)

Consider enabling the “Optimize for Ad-Hoc workloads” server option on SQL Server 2008 or later. This option is used to improve the efficiency of the plan cache for workloads that contain many single use ad hoc batches.

With this option enabled, when a query executes the first time, only the query hash is stored in cache. If the same plan is reused, then it is deemed fit for storing the entire plan in cache. While this effectively adds a small delay in a second execution, the benefits outweigh the cost: if a query is executed countless times, the query plan only gets stored in cache on the second execution which is probably not a major drawback, but if not, memory is saved by not caching query plans that won't get used.

In certain scenarios, Dynamics may experience degraded performance and contribute to memory pressure in the database engine due to the number of cached single use plans. This usually manifests itself with complaints that the server performance degrades over time until SQL Server is restarted. In these cases, enabling “Optimize for Ad-Hoc workloads” can help. Although not a very common occurrence in the context of Microsoft Dynamics NAV, it can still happen. Therefore, if memory pressure becomes an issue, consider enabling this setting.

References and further reading material:

- [Optimize for ad hoc workloads Option](#)

SQL Server Files

Physically isolate Data files from Log files. Combining heterogeneous workloads (workloads with very different I/O and latency requirements) can have negative effects on overall performance.

References and further reading material:

- [Storage Best Practices for SQL Server](#)
- [Storage Top 10 Best Practices](#)
- [Separate database and transaction log files on different drives for optimal performance and disaster recovery](#)

In addition, it is strongly recommended to **place the database files and backups on separate devices**. Otherwise, if the device that contains the database fails, your backups will be unavailable as well. Furthermore, placing the data (meaning Data and Log files) and backups on isolated devices optimizes the I/O performance for both production use of the database and writing backups.

Use the Instant File Initialization feature as this will allow a boost in performance, because when extending or creating a data file (does not apply to log files), it will not be

zeroed right away, saving time and I/O cycles. **This can be accomplished by granting the "Perform volume maintenance tasks" permission to the SQL Server account**, which is set in the security policy console (*secpol.msc*).

This permission comes with a small security risk, because by not zeroing out the existing space, when deleting data for example, there is a possibility that data could still be read, even though it has been "deleted", until some other data writes on that specific area of the data file. However, the performance benefits often outweighs the security risk and hence this option should be considered when safe to do so.

References and further reading material:

- [How to tell if you have instant initialization enabled?](#)
- [How and Why to Enable Instant File Initialization](#)

When setting AUTOGROW for Data and Log files, keep in mind that it might be preferred to set it in Megabytes instead of Percentage, to allow better control on the growth ratio, as percentage is an ever-growing amount. This is even more critical when Instant File Initialization is not in use, as long I/O might become a bottleneck. Keep in mind that transaction logs cannot leverage Instant File Initialization, so extended log growth times are especially critical. As a rule, do not set any AUTOGROW value above 1024MB.

Also, if you grow your database by small increments (or if you grow it and then shrink it) you can end up with disk fragmentation. Disk fragmentation can cause performance issues in some circumstances. So, always try to pre-size data and log files and **do not rely on AUTOGROW for growth control**, instead manage the growth of these files manually and gain more control over I/O and storage requirements. You may leave AUTOGROW ON for safety reasons, but use alerts or monitoring programs to monitor file sizes and proactively manage the growth of the data files. This helps you avoid fragmentation and permits you to shift these maintenance activities to non-peak hours.

References and further reading material:

- [Considerations for the "autogrow" and "autoshrink" settings in](#)
- [SQL Server SQL Server reports Msg 5144 and 5145 for long or failed autogrow of databases and transaction log files](#)

Optimize TempDB file layout by creating one data file for each two processor cores, and put them on separate LUNs if possible (although this is not a requisite), up to 8 files (mileage may vary). The TempDB system database is a global resource available to all users connected to the instance of SQL Server and holds all temporary tables and temporary stored procedures. It also fills any other temporary storage requirements such as worktables and workfiles generated by SQL Server. The configuration of TempDB can affect your SQL Server performance.

Having multiple TempDB data files can reduce contention and improve performance on active systems. This is because there will be one or more SQL Server GAM (Global Allocation Map) and SGAM (Shared GAM) pages for each file. The GAM/SGAM are the main point of contention in this type of issue.

Dividing TempDB into multiple data files of equal size provides a high degree of parallel efficiency in operations that use TempDB. These multiple files do not necessarily need to be on different disks or spindles unless you are also encountering I/O bottlenecks as well.

One disadvantage of having too many TempDB files is that every object in TempDB will have multiple SQL Server IAM (Index Allocation Map) pages. In addition, there will be more switching costs as objects are accessed as well as more managing overhead. On very large systems, 8 TempDB data files may be sufficient, but reconsider this based on the workload. Be aware that if one of the data files is larger than the others, this mechanism will not be effective, so regularly check to see if growth (via AUTOGROW) has broken the proportional fill.

Trace Flag 1118 will allow uniform extent allocations (i.e. “full-mixed”) instead of “mixed” extent allocations. **Consider enabling this trace** if the TempDB is in high demand and latch contention is observed on wait resource 2:1:1. This is Database #2 (i.e. TempDB), File #1, Page #1, i.e. the first PFS (Page Free Space) page. Also monitor 2:1:3 (the first SGAM page). Otherwise, test the trace flag impact on your workload.

References and further reading material:

- [*FIX: Concurrency enhancements for the TempDB database*](#)
- [*Optimizing TempDB Performance*](#)
- [*Capacity Planning for tempdb*](#)
- [*TEMPDB Capacity Planning and Concurrency Considerations for Index Create and Rebuild*](#)
- [*Working with TempDB in SQL Server® 2005*](#)
- [*SQL Server Urban Legends Discussed SQL Server \(2005 and 2008\) Trace Flag 1118 \(-T1118\) Usage*](#)
- [*Misconceptions around TF 1118*](#)
- [*Inside The Storage Engine: GAM, SGAM, PFS and other allocation maps*](#)
- [*Inside the Storage Engine: Anatomy of an extent*](#)

Data files should be of equal size within each Filegroup, as SQL Server uses a proportional fill algorithm that favors allocations in files with more free space. You may recognize this recommendation as directed at the TempDB, but it applies to all user databases also.

References and further reading material:

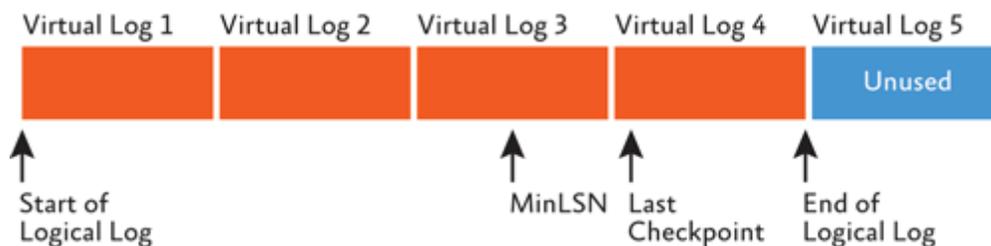
- [File and Filegroup Fill Strategy](#)
- [Adding and Deleting Data and Transaction Log Files](#)

Pre-size log files as needed and do not rely on AUTOGROW. The growth increment of your transaction log must be large enough to stay ahead of the needs of your transaction units. Even with autogrow turned on, you can receive a message that the transaction log is full, if it cannot grow fast enough to satisfy the needs of your query.

A good pointer to properly size a transaction log is monitoring the amount of log occupied during:

- The time required to execute a full backup, because log backups cannot occur until it finishes.
- The time required for the largest index maintenance operations.
- The time required to execute the largest batch in a database.

Beware of VLF (Virtual Log File) issues that can influence SQL Server operations: backups, recovery, DML or even replication. SQL Server allocates log space in logical sections called Virtual Log Files (VLFs) within each physical log file.



Adjust your transaction log files to their needed size in advance. Each VLF spawned should not exceed 512MB, so calculate the needed autogrow setting based on this guideline and the VLF creation methods, outlined below. VLF creation is achieved according to the following method:

- Growth in chunks less than 64MB will spawn 4 VLFs.
- Growth in chunks of 64MB and up to 1GB will spawn 8 VLFs.
- Growth in chunks of larger than 1GB will spawn 16 VLFs.

References and further reading material:

- [Virtual Log Files](#)
- [Transaction Log Physical Architecture](#)
- [8 Steps to better Transaction Log throughput](#)
- [Optimizing Transaction Log Performance](#)
- [Managing the Size of the Transaction Log File](#)
- [Recover from a full transaction log in a SQL Server Database](#)
- [Considerations for the "autogrow" and "autoshrink" settings in SQL Server](#)
- [SQL Server and Log File usage](#)
- [Understanding Logging and Recovery in SQL Server](#)
- [How a log file structure can affect database recovery time](#)
- [Can log files growth affect DML? Can log files growth affect DML? Post-CU Update](#)
- [VLFs revisited](#)
- [FIX: Slow performance when you recover a database if there are many VLFs inside the transaction log in SQL Server 2005, in SQL Server 2008 or in SQL Server 2008 R2](#)

Database Recommended Settings

If you are running SQL Server 2008 or above and Microsoft Dynamics NAV 2013, the Database Compatibility level must be set to 100 or higher in order for Microsoft Dynamics NAV to use the OPTIMIZE FOR UNKNOWN query hint, to help avoid parameter sniffing. Database Compatibility level for each database should be set to the same version as the SQL Server Engine where the Microsoft Dynamics NAV database is running.

References and further reading material:

- [Setting Database Options](#)
- [OPTIMIZE FOR UNKNOWN – a little known SQL Server 2008 feature](#)

Consider enabling Snapshot Isolation if other applications access Microsoft Dynamics NAV databases. Since the release of SQL Server 2005, a new isolation level was introduced called **Read Committed Snapshot**. If READ_COMMITTED_SNAPSHOT is set to ON, the Database Engine uses row versioning to present each statement with a transactionally consistent snapshot of the data as it existed at the start of the statement. Locks are not used to protect the data from updates by other transactions. This means readers and writers will receive different versions of the record, so they do not block each other while maintaining transactional consistency. These row versions are supported by the Version Store, which is kept in TempDB.

From the Microsoft Dynamics NAV application perspective this is not very useful as Microsoft Dynamics NAV sends isolation level hints with most statements which will override RCSI, but will have a benefit for outside applications accessing records within the Microsoft Dynamics NAV production database, such as, reports, views, and secondary applications, like SQL Server Reporting Services or Integration Services ETL (Extract-Transform-Load) processes.

References and further reading material:

- [Snapshot Isolation in SQL Server](#)
- [Enabling Row Versioning-Based Isolation Levels](#)
- [SET TRANSACTION ISOLATION LEVEL \(Transact-SQL\)](#)

Database options such as **“AUTOSHRINK”** and **“AUTOCLOSE”** should be set to OFF. If set to ON, these options might create unnecessary I/O overhead, file fragmentation and severely affect overall performance.

References and further reading material:

- [Setting Database Options](#)
- [Considerations for the "autogrow" and "autoshrink" settings in SQL Server](#)

Database options such as **“AUTO_CREATE_STATISTICS”** and **“AUTO_UPDATE_STATISTICS”** should be set to ON (default), unless some very specific scenario advises otherwise.

The AUTO_CREATE_STATISTICS option does not determine whether statistics are created for indexes, although the query optimizer creates statistics for indexes on tables or views when the index is created. These statistics are created on the key columns of the index. If the index is a filtered index, it creates filtered statistics on the same subset of rows specified for the filtered index.

When AUTO_CREATE_STATISTICS is set to ON, if the query optimizer does not find any statistics on columns used in a predicate, it will create the missing single-column statistics. These single-column statistics are created only on columns that are not already the first column of an existing statistics object. These events can be observed in a SQL Profiler trace as a “Missing Column Statistics” warning.

When AUTO_UPDATE_STATISTICS is ON, the query optimizer automatically updates this statistical information periodically within certain thresholds:

- When the number of rows in the table was 500 or less at the time statistics were created, update every 500 modifications.
- When the number of rows in the table was above 500 at the time statistics were created, update every 500 + 20% of modifications.

It is essential to update the statistics on a regular basis because out-of-date statistics can severely influence query performance. In some circumstances, statistical sampling will not be able to accurately characterize the data in a table. In that situation, update the statistics manually where you control the amount of data that is sampled.

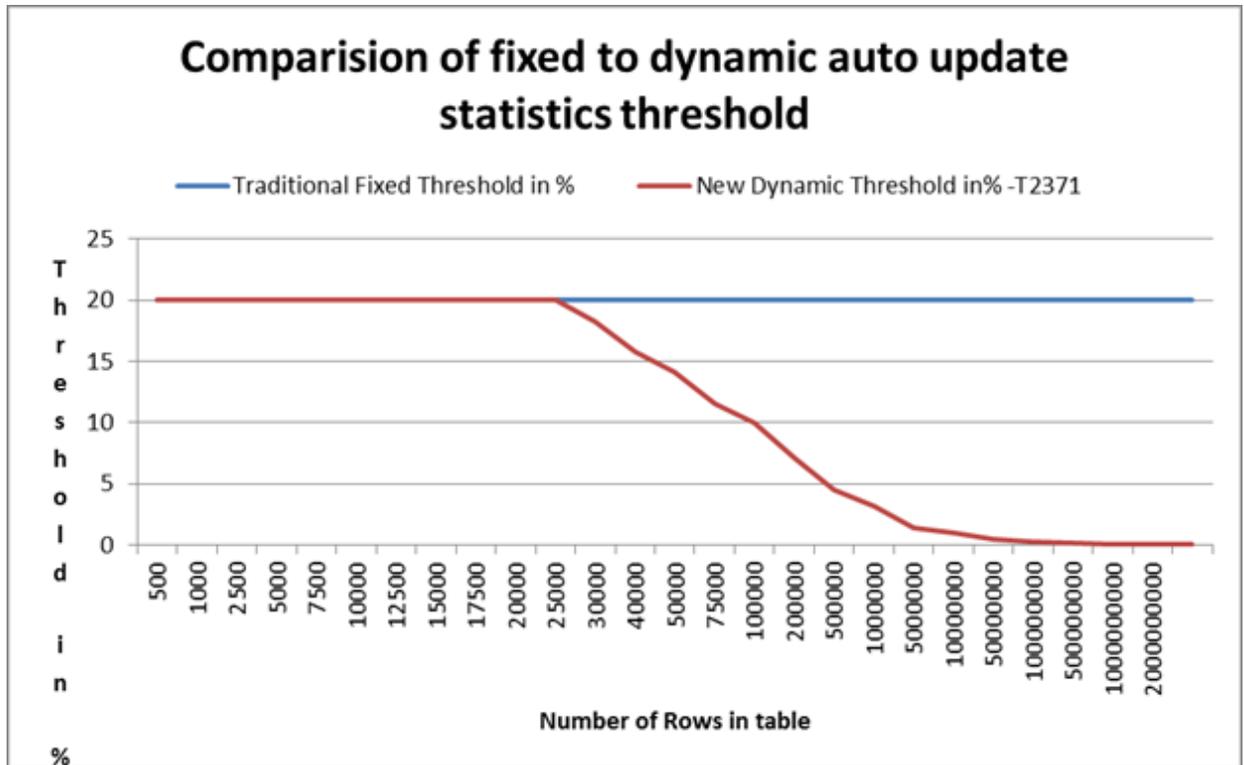
This is very important with C/AL **FINDFIRST**, **FINDLAST**, **FIND('-')**, **FIND('+')**, and **FINDSET** for **[Entry No.]**, because if the statistic on **[Entry No.]** is out of date, it will take longer for SQL Server to find the first or last value. This means it will lock the resource longer, with negative impact on system concurrency and possibly resulting in Lock timeouts.

References and further reading material:

- [*Setting Database Options*](#)
- [*How Online Index Operations Work*](#)
- [*TempDB and Index Creation*](#)
- [*Statistics Used by the Query Optimizer in Microsoft SQL Server 2008*](#)
- [*The auto update statistics option, the auto create statistics option, and the Parallelism setting are turned off in the SQL Server database instance that hosts the BizTalk Server BizTalkMsgBoxDB database*](#)
- [*Auto update statistics and auto create statistics - should you leave them on and/or turn them on?*](#)

If the Microsoft Dynamics NAV database is on a dedicated SQL Server 2008 R2 SP1 or above, consider enabling trace flag 2371. This global trace flag changes the fixed rate thresholds for update statistics into a dynamic percentage rate. The higher the number of rows in a table, the lower the threshold will become to trigger an automatic update of the statistics.

On tables such as **Value Entry**, **Item Ledger Entry** or **G/L Entry**, that can have 10s of millions of records, a small percentage of data changes, like in column **[Entry No.]**, can have a material effect on the overall data distribution in the associated statistic, causing inefficient query plans long before the 5% threshold is reached, resulting in degraded performance.



References and further reading material:

- [Changes to automatic update statistics in SQL Server – traceflag 2371](#)

Set **PAGE_VERIFY** database option to “Page Checksum”. If you migrate old SQL Server 2000 databases to later versions of SQL Server, be sure to manually update the PAGE_VERIFY setting for the database to “Page Checksum”, regardless of the database compatibility level.

References and further reading material:

- [Set the PAGE_VERIFY Database Option to CHECKSUM](#)

Database **Recovery model** should be set to Full, unless point in time recovery is not needed or is otherwise a design spec. If set to Full, it is imperative to make log backups with the appropriate scheduling.

SQL Server Maintenance

Even with "AUTO_UPDATE_STATISTICS" enabled, if trace flag 2371 is not enabled, it is still strongly recommended to **run a periodic SQL Maintenance Job to update statistics**. We recommend creating a SQL Job that runs daily or weekly (depending on transaction volume), during off peak hours, to update all statistics where data has changed.

**Do Not:**

Do not use "FULLSCAN" as this is a waste of time. It will be overwritten the second the statistic is "auto updated."

References and further reading material:

- [Adaptive Index Defrag](#)
- [Statistics Used by the Query Optimizer in Microsoft SQL Server 2008](#)

Implement maintenance plans for clean-up tasks, rebuilding or reorganizing indexes, updating statistics and integrity checking. If needed, have the maintenance plans perform Transaction Log backups and Database backups as well. These maintenance plans are not necessarily Maintenance Plan objects, but can be a customized set of scripts and jobs that perform all the required tasks.

Remember to take backups with checksum and validate backups regularly. These maintenance plans can be customized jobs to perform these tasks, or the Maintenance Plan DTSx packages available in SQL Server.

References and further reading material:

- [How to: Create a Maintenance Plan](#)

Execute database integrity checks with DBCC CHECKDB regularly. The frequency in which DBCC CHECKDB is run against any particular database depends largely on the individual business needs and the importance of the information in the database. However, at a minimum, DBCC CHECKDB should be run against all production databases at least once a week. In addition, the results should be reviewed as soon as possible after execution completes in order to identify and resolve any errors before they become critical. DBCC CHECKDB in SQL Server 2005 and later versions use an internal database snapshot to perform the checks in order to avoid blocking and concurrency problems. However, because SQL Server 2005 performs much more extensive checks than earlier versions, it may take considerably longer to complete. Thus, Microsoft recommends that the PHYSICAL_ONLY option be used for frequent checks on production databases.

References and further reading material:

- [SQL Server® 2008 R2 Books Online: DBCC CHECKDB \(Transact-SQL\)](#)

Lesson 5: Microsoft Dynamics NAV Settings

The following describes two of SQL Server's isolation levels:

- **SERIALIZABLE** is the most restrictive isolation level. When used, the phantom reads cannot occur. It blocks other transactions from updating or inserting any rows that would qualify for any of the statements executed by the current transaction.
- **REPEATABLE READ** does not allow dirty reads and non-repeatable reads. This means that shared locks are placed on all data read by each statement in the transaction and are held until the transaction completes. Therefore, statements cannot read data that has been modified but not yet committed by other transactions and that no other transactions can modify data that has been read by the current transaction until the current transaction completes.

Earlier versions of Microsoft Dynamics NAV (i.e. before Microsoft Dynamics NAV 2013) used to the SERIALIZABLE isolation level by default when updating data. It was possible to manually enable the REPEATABLE READ isolation level for certain builds of Microsoft Dynamics NAV 5.00 SP1 and Microsoft Dynamics NAV 2009 SP1/R2. However, most customers were using the default SERIALIZABLE isolation level.

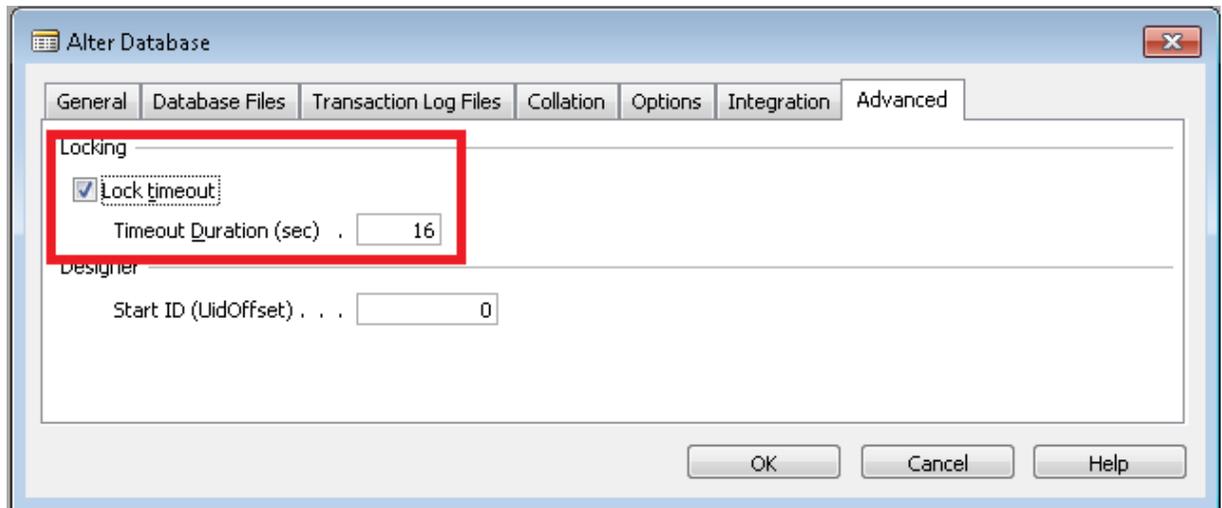
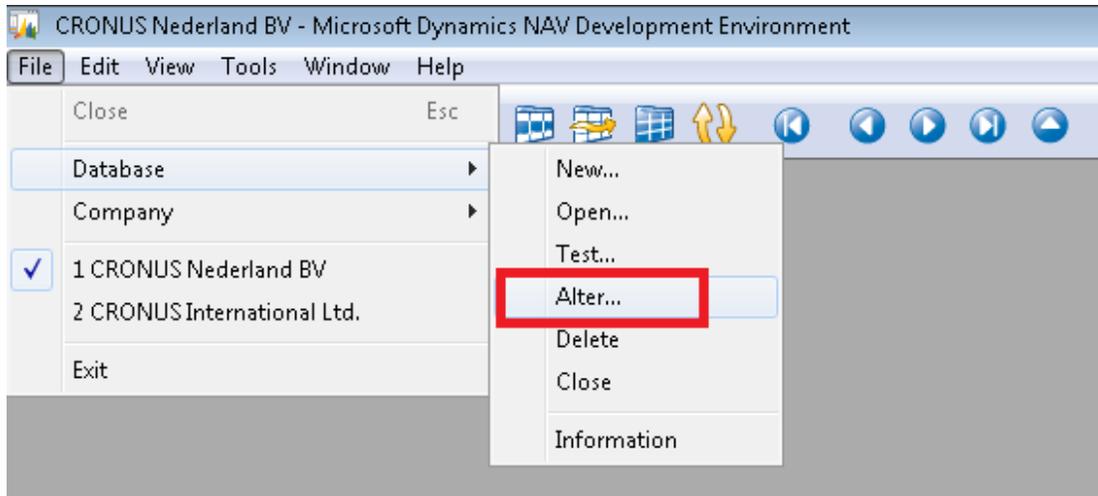
Microsoft Dynamics NAV 2013 uses the REPEATABLE READ isolation level by default. It is not possible to manually switch to the SERIALIZABLE isolation level, as this would reduce database performance for no practical benefit.

References and further reading material:

- [Isolation Levels in the Database Engine](#)
- [Microsoft Dynamics NAV Changes by Version](#)
- [SET TRANSACTION ISOLATION LEVEL \(Transact-SQL\)](#)

It is generally recommended not to use the Lock timeout setting, or to set it between 10 and 20 seconds, if needed. This setting essentially means that if a user attempts to access a resource and it is locked by another user, it will wait however many seconds specified here and try again before issuing an error message. Having a very high value for the static lock time-out can cause its own issues. When this configuration is set to a high value, usually there is an underlying performance issue.

However, there are specific circumstances where this does need to be set higher, so it is really up to the customer or partners discretion to determine the proper value in such circumstances. This setting can be found under *File -> Database -> Alter*, on the *Advanced* tab, as shown below.



In earlier versions of Microsoft Dynamics NAV, i.e. before NAV 2013, it was possible to configure the “Always Rowlock” setting to try to force SQL Server to use row level locking. This setting was a holdover from SQL Server 2000 and 32-bit architecture. With newer versions of SQL Server, the recommendation was to disabled this Microsoft Dynamics NAV setting to free up lock memory in the database engine without any loss in performance. In some benchmark tests, not having this setting enabled actually increases concurrency and performance.

The “Always Rowlock” setting has been removed from Microsoft Dynamics NAV 2013. It would no longer be useful as no versions of SQL Server prior to SQL Server 2008 is supported.

References and further reading material:

- [Microsoft Dynamics NAV/SQL Server Configuration Recommendations](#)

Lesson 6: The Microsoft Dynamics NAV 2013 Security Model

Microsoft Dynamics NAV 2013 Users and Credential Types

Microsoft Dynamics NAV 2013 supports four credential authorization mechanisms for Microsoft Dynamics NAV users. When you create a user, you provide different information depending on the credential type that you are using in the current Microsoft Dynamics NAV Server instance. You specify which credential type is used for a particular Microsoft Dynamics NAV Server instance by configuring the relevant RoleTailored client configuration file and Microsoft Dynamics NAV Server instance configuration.

**Note:**

All users of a Microsoft Dynamics NAV Server instance must be using the same credential type.

Credential Types

Microsoft Dynamics NAV supports the following credential types:

- **Windows** - With this credential type, users are authenticated using their Windows credentials (Active Directory, or local workgroup, or the local computer's users). Before you create a Windows user in the RoleTailored client, there must already be a corresponding user in Windows. In Microsoft Dynamics NAV 2009, all RoleTailored client users were Windows users. Because they are authenticated through Windows, Windows users are not prompted for credentials when they start the RoleTailored client.
- **Username** - With this setting, the user is prompted for username/password credentials when starting the RoleTailored client. These credentials are then validated against Windows authentication by Microsoft Dynamics NAV Server. There must already be a corresponding user in Windows. Security certificates are required to protect the passing of credentials across a wide-area network. This setting should be used when the Microsoft Dynamics NAV Server computer is part of an authenticating Active Directory domain, but the computer where the RoleTailored client is installed is not.
- **NavUserPassword** - With this setting, authentication is managed by Microsoft Dynamics NAV Server but is not based on Windows users or Active Directory. The user is prompted for username/password credentials when they start the RoleTailored client. The credentials are then validated by an external mechanism. Security certificates are required to protect the passing of credentials. This mode is intended for hosted environments, for example, where Microsoft Dynamics NAV is implemented in Azure.

- **AccessControlService** - With this setting, Microsoft Dynamics NAV relies on Windows Azure Access Control Service (ACS) for user authentication services. ACS is a cloud-based service that provides user authentication and authorization for web applications and services. ACS integrates with standards-based identity providers, including enterprise directories such as Active Directory, and web identities such as Windows Live ID, Google, Yahoo!, and Facebook.

Configuring the Credential Type

The RoleTailored client and Microsoft Dynamics NAV Server must be configured to use the same credential type.

1. Edit the RoleTailored client configuration file for each relevant user.

A separate instance of the ClientUserSettings.config file is maintained for each client user. You must edit the configuration for each instance of the file. The default location for this file is

C:\Users\<<username>\AppData\Roaming\Microsoft\Microsoft Dynamics NAV\70, (or %appdata%\Microsoft\Microsoft Dynamics NAV\70)

In the ClientUserSettings.config file:

- Find the **ClientServicesCredentialType** parameter and change the value to one of the options listed above, e.g. Windows.
 - Save ClientUserSettings.config and restart the RoleTailored client.
2. There are two ways to edit the configuration for the Microsoft Dynamics NAV Server instance.
 - Using the Microsoft Dynamics NAV Server Administration tool, find the Credential Type and select the desired value from the drop down list.



Note:

In the Microsoft Dynamics NAV Server Administration tool, the parameter is named **Credential Type** and is on the **General** tab.

- Edit the CustomSettings.config file for the NAV Server instance. This can normally be found in the following folder:

%programfiles%\Microsoft Dynamics NAV\70\Service

Change the **ClientServicesCredentialType** parameter to one of the valid listed above, e.g. Windows.

Restart the Microsoft Dynamics NAV Server instance after making any changes to the configuration options. You can do this using any of the following methods:

- The Microsoft Dynamics NAV Server Administration tool.
- The Services tool in Windows Control Panel.
- The Microsoft Dynamics NAV PowerShell API.

Creating new users for Microsoft Dynamics NAV

To create a new user for the RoleTailored client, use the **Users** window in the RoleTailored client.

It is not necessary to create user logins in SQL Server before you create Microsoft Dynamics NAV users. There is no longer any requirement to Synchronize Microsoft Dynamics NAV logins with SQL Server logins.

As mentioned above, Microsoft Dynamics NAV 2013 supports the following credential authorization mechanisms for Microsoft Dynamics NAV users:

- Windows
- Username
- NavUserPassword
- AccessControlService

To create a new user for Windows authentication

1. In the RTC navigate to the following Page:
CRONUS International Ltd./Departments/Administration/IT
Administration/General/Users
2. In the **Users** Page, on the **Home** tab, select **New**.
The **New - Users Card** window opens.



Note:

Any fields not mentioned in this procedure are not relevant for Windows authentication.

3. In the **User Name** field, type a unique, short name to identify the user.
4. In the **Full Name** field, type the user's full name (first name and last name).
5. In the **License Type** field, select one of the available license types.



More:

For more information, see:

License Types (to come)

6. In the **State** field, specify whether the user's access should be **Enabled** or **Disabled**.
7. To set a time limit on the user's access, select a date in the **Expiry Date** field.
8. In the **Windows User Name** field, type the name of a valid Active Directory user, using the format **domain\username**. Optionally, click **AssistEdit**, select **Allow for the Session**, and then, in the **Active Directory Select User or Group** dialog box, identify a Windows user.
9. Select the **User Permission Sets** FastTab to define permission sets for the user. Select the first row under Permission Sets in and then select a permission set. Select additional permission sets as needed.
10. Add any Notes or Links for the user in the respective FactBoxes.
11. Click **OK** to close the window.

To update the settings for an already existing user in the **Users** window, select the user and then, on the **Home** tab, select **Edit**.

To create a new user for Username authentication

1. In the RTC navigate to the following Page:
CRONUS International Ltd./Departments/Administration/IT
Administration/General/Users
2. In the **Users** Page, on the **Home** tab, click **New**.
The **New - Users Card** window opens.



Note:

Any fields not mentioned in this procedure are not relevant for Username authentication.

3. In the **User Name** field, type a unique, short name to identify the user.
4. In the **Full Name** field, type the user's full name (first name and last name).
5. In the **License Type** field, select one of the available license types.



More:

For more information, see:

License Types (to come)

6. In the **State** field, specify whether the user's access should be **Enabled** or **Disabled**.

7. If you want to set a time limit on the user's access, select a date in the **Expiry Date** field.
8. In the **Windows User Name** field, type the name of a valid Active Directory user, using the format **domain\username**. Optionally, click **AssistEdit**, select **Allow for the Session**, and then, in the **Active Directory Select User or Group** dialog box, identify a Windows user.
9. Select the **User Permission Sets** FastTab to define permission sets for the user. Select the first row under Permission Sets in and then select a permission set. Select additional permission sets as needed.
10. Add any Notes or Links for the user in the respective FactBoxes.
11. Click **OK** to close the window.

To update the settings for an already existing user in the **Users** window, select the user and then, on the **Home** tab, select **Edit**.

To create a new user for NAVUserPassword authentication

1. In the RTC navigate to the following Page:
CRONUS International Ltd./Departments/Administration/IT
Administration/General/Users
2. In the **Users** Page, on the **Home** tab, click **New**.
The **New - Users Card** window opens.

**Note:**

Any fields not mentioned in this procedure are not relevant for NavUserPassword authentication.

3. In the **User Name** field, type a unique, short name to identify the user.
This will be the username that the user uses, in combination with a password, to access the RoleTailored client.
4. In the **Full Name** field, type the user's full name (first name and last name).
5. In the **License Type** field, select one of the available license types.

**More:**

For more information, see:
License Types (to come)

6. In the **State** field, specify whether the user's access should be **Enabled** or **Disabled**.

7. If you want to set a time limit on the user's access, select a date in the **Expiry Date** field.
8. Under **NAV Username/Password Auth**, enter a **User Name**.
9. Specify a **Password** for the user.
Alternatively, you can click **AssistEdit** to open a separate dialog box where you can specify, and then confirm, a password.
10. If you want to enable the user to change the password, select **Password To Be Changed**.
The first time that the user logs on, a prompt will appear prompting the user to change the password.
11. Select the **User Permission Sets** FastTab to define permission sets for the user. Select the first row under Permission Sets in and then select a permission set. Select additional permission sets as needed.
12. Add any Notes or Links for the user in the respective FactBoxes.
13. Click **OK** to close the window.

To update the settings for an already existing user in the **Users** window, select the user and then, on the **Home** tab, click **Edit**.

To create a new user for AccessControlService authentication

1. In the RTC navigate to the following Page:
CRONUS International Ltd./Departments/Administration/IT
Administration/General/Users
2. In the Users Page, on the Home tab, click New
The **New - Users Card** window opens.



Note:

Any fields not mentioned in this procedure are not relevant for AccessControlService authentication. AccessControlService authentication is relevant only for Microsoft Dynamics NAV solutions that are deployed on Azure.

3. In the **User Name** field, type a unique, short name to identify the user.
This will be the username that the user uses, in combination with a password, to access the RoleTailored client.
4. In the **Full Name** field, type the user's full name (first name and last name).
5. In the **License Type** field, select one of the available license types.

**More:**

For more information, see:

License Types (to come)

6. In the **State** field, specify whether the user's access should be **Enabled** or **Disabled**.
7. If you want to set a time limit on the user's access, select a date in the **Expiry Date** field.
8. In the **Authentication** section, next to **ACS Access Status**, click on the word **Disabled**, or select the word and press **Enter**.
This opens an **Edit -User ACS Setup** dialog box.
9. In the **Edit -User ACS Setup** dialog box, click **Generate Auth Key** on the ribbon.
A string value is automatically inserted into the **Authentication Key** field.
10. Save the **Authentication Key** value to a text file or another location where you can find it later. After you configure you deployment for ACS, and configure your Microsoft Dynamics NAV components for ACS, send this value, along with the **User Name** value, to the actual user and direct the user to provide these values when logging on with a Microsoft Dynamics NAV client.
11. Click **OK** to exit the **Edit -User ACS Setup** dialog box. The ACS Access Status is now **Pending**. It will change to **Active** after the user successfully logs in.
12. Select the **User Permission Sets** FastTab to define permission sets for the user. Select the first row under Permission Sets in and then select a permission set. Select additional permission sets as needed.
13. Add any Notes or Links for the user in the respective FactBoxes.
14. Click **OK** to close the window.

Database Logins

**Note:**

Database login are only valid for connecting to the database from the Microsoft Dynamics NAV Development Environment.

Users are given a database login when they have their own user ID and password in Microsoft Dynamics NAV. The user must enter the user ID and password to access the database.

How Database Logins Work

Users must also have a login on SQL Server. SQL Server has its own authentication of the user's ID and password. SQL Server does this by checking whether a SQL Server login with this user's ID and password has been created.

This login must first be created by a SQL Server administrator, with a SQL Server tool. If a SQL Server login has not been set up, authentication fails and the user receives an error.

The user is granted access to the server after his login has been authenticated. Database security then validates the user's permissions by checking the database user accounts on the server. The permissions that the user has been granted to the various objects within the database, such as tables, are determined by the information contained in the user's database user account. This account also contains information about any additional permissions that the user may have been granted to alter the database itself.

Other Security Considerations

The following list contains some important things to consider when setting up your Microsoft Dynamics NAV security system.

- The Microsoft Dynamics NAV security system is initiated when you create the first login. Until you create the first login, any user can have full access to carry out any transaction in a Microsoft Dynamics NAV database. Therefore, the first login that you create must be a superuser. The superuser must own and administer all access to the database from within Microsoft Dynamics NAV. We recommend that the user who administers security in Microsoft Dynamics NAV be a superuser.
- One of the first things that the superuser should do is create logins for the other users that will have access to the database and grant them the appropriate permissions.
- In Microsoft Dynamics NAV, a table can contain a FlowField, which generates sums based on values that are stored in another table. When using a FlowField, a user must have permission to read both tables or they will not be allowed to read the first table