

Microsoft Business Solutions

Tuning Navision for better performance

Revision 5

Created May 21, 2004

SUMMARY

This article provides a list of items to check to ensure that a Navision implementation meets all requirements for performance optimization.

You can write for each item in a specific (grayed) area the current or planned implementation settings. Comparison with recommendations is made easier.

The article provides generic information. Recommendations may vary upon specific constraints.

The article contains the following sections:

SUMMARY	2
DISCLAIMER	2
MORE INFORMATION	3
Server setup checklist	3
Standard Editions vs. Enterprise Editions	3
Server RAID technology	4
Server disks configuration	5
Sever performance counters to monitor	6
Clients setup checklist	9
Windows 2000 Professional vs. Windows XP Professional	9
Client disks configuration	9
Client tools to run	9
SQL database setup checklist	10
SQL Database Files (does not apply with stripped RAID strategies)	10
SQL indexes rebuilt	11
SQL Indexes FillFactor	11
SQL Statistics	11
Customisation code review checklist	13
Deadlocks investigation with SQL Server	16
1. Turn on the trace flags in SQL Server	16
2. Analyze each deadlock encountered to know exactly what Navision processes are involved	17
3. Correct the Navision C/AL code and/or report the error to Microsoft	17
Known issues in Navision 3.70	18
Codeunit 12 – application posting process	18
Codeunit 80 / codeunit 90 and codeunit 5704 / codeunit 5705	18
Codeunit 80 / codeunit 90 and codeunit 226 / codeunit 227	18
REFERENCES	19

DISCLAIMER

This document is provided "AS IS" with no warranties, and confers no rights.

Use of included script samples are subject to the terms specified at

<http://www.microsoft.com/info/copyright.htm>.

MORE INFORMATION

Server setup checklist

Standard Editions vs. Enterprise Editions

	RAM	Max. CPUs
Windows 2000 Windows 2003 Standard Ed. <i>with</i> SQL Server 2000 Ed.	<ul style="list-style-type: none"> 4GB: 2GB allocated to SQL Server and other user applications. 	4
Windows 2000 Windows 2003 Standard Ed. <i>with</i> SQL Server 2000 Enterprise Ed.	<ul style="list-style-type: none"> 4GB: enable AWE memory in SQL Server to allocate up to 4GB to SQL Server. 	4
Windows 2000 Advanced Server <i>with</i> SQL Server 2000 Enterprise Ed.	<ul style="list-style-type: none"> 4GB: use the /3GB switch in the Boot.ini file to allocate up to 3GB to SQL Server and other user applications. Enable AWE memory in SQL Server. 8GB when used with processors supporting Physical Address Extensions (PAE): use the /3GB /PAE switches in the Boot.ini file to enable PAE and allocate up to 7GB to SQL Server and other user applications. Enable AWE memory in SQL Server. 	8
Windows 2003 Enterprise Ed. <i>with</i> SQL Server 2000 Enterprise Ed.	<ul style="list-style-type: none"> 32GB when used with processors supporting Physical Address Extensions (PAE): use the /PAE switch in the Boot.ini file to enable PAE. Enable AWE memory in SQL Server. 	8

Customer's Server Setting		
Operating System (Tick the version used)	Windows 2000	
	Windows 2003 Standard Ed.	
	Windows 2000 Advanced Server	
	Windows 2003 Enterprise Ed.	
SQL Server (Tick the version used)	SQL Server 2000 Ed.	
	SQL Server 2000 Enterprise Ed.	
RAM		GB
/3GB (/PAE) switch (Tick the current setting)	Yes	
	No	
AWE Memory enabled (Tick the current setting)	Yes	
	No	
Number of CPUs		

Server RAID technology

RAID	Description	Advantages Disadvantages	Recommended System	Customer's RAID Setting (Tick if used)	RAID 10 Recommended
0	Data are stripped across several disks	Write/Read transactions are faster on the partition Not Fault Tolerant	Not to be used to store critical information		
1	Data are duplicated on mirror disks	Disk failure can be recovered with no system interruptions. Duplexing(1) offers higher performance Write transactions are duplicated – read transactions are faster due to concurrent use of all functioning disks	ERP systems (with numerous write and read transactions)		
1E	Data are stripped and mirrored	Requires lesser number of disks than RAID 10 but offers the same level of security Write transactions are duplicated – concurrent use of all functioning disks limited	ERP systems (with numerous write and read transactions)		
5	Parity bits are calculated for data written on striped disks	Require only 1 extra disk on top of the data disks Write transactions require the calculation of a parity bit	File/Application /Internet Servers - Archive systems (with numerous read transactions, few write transactions)		

(1) 1 Disk controller per disk

Server disks configuration

	Expected Files Size	Recommended Logical Drive Letter	Customer's Logical Drive Letter
Operating System	Windows 2000 Server 240-520 MB Windows 2000 Advanced Server 750-950 MB SQL Server 2000 (Standard & Enterprise) 100-250 MB Windows 2003 Standard / Enterprise Ed. 1.5 GB	C	
Swap File	1.5 to 2.5 times the amount of RAM	C	
SQL executables	170MB	R or S	
SQL Temp Db Database	9GB	F through H	
Transaction Log File	3 times the database size	T through V	
Database File		I through P	

Sever performance counters to monitor

Object Name	Counter Name	Instances	Best Values	Recommendation (Best Values not met)
Memory	Available Mbytes	SQL Server TS Servers	>5MB	Add more memory Reserve less memory for SQL Server
	Pages/sec	SQL Server TS Servers	<25	Add more memory Reserve less memory for SQL Server
Physical Disk	Avg Disk Read Queue length	SQL Server Disks	<2	Change disk system
	Avg Disk Write Queue length	SQL Server Disks	<2	Change disk system
Processor	% Processor time	SQL Server TS Servers	0-80	Add more CPUs
System	Processor Queue Length	SQL Server TS Servers	<2	Add more CPUs
	Context Switches/sec	SQL Server TS Servers (Multi Processors)	<8000	Set Affinity Mask
Network Interface	Output queue length	SQL Server TS Servers	<2	Increase network capacity
SQL Server Access Methods	Full Scans/sec	SQL Server		Review Navision C/AL Code
	Page Splits/sec	SQL Server	0	Defrag SQL Server Indexes Review Navision C/AL Keys
SQL Server Buffer Manager	Buffer Cache Hit Ratio	SQL Server	>90	Add more memory
SQL Server Databases	Log Growths	SQL Server	0 (during peak times)	Increase and set the size of the transaction log
SQL Server General Statistics	User Connections	SQL Server		
SQL Server Locks	Lock Requests/sec	SQL Server		Review Navision C/AL Code
	Lock Wait/sec	SQL Server		Review Navision C/AL Code

Object Name	Counter Name	Instances	Customer's Counters' Values
Memory	Available Mbytes	Database Servers	Average Max Value Min Value
	Available Mbytes	TS Servers	Average Max Value Min Value
	Pages/sec	Database Server	Average Max Value Min Value
	Pages/sec	TS Servers	Average Max Value Min Value
Physical Disk	Avg Disk Read Queue length	Swap File Disk	Average Max Value Min Value
	Avg Disk Read Queue length	Transaction Log File Disk	Average Max Value Min Value
	Avg Disk Read Queue length	Database File Disk	Average Max Value Min Value
	Avg Disk Write Queue length	Swap File Disk	Average Max Value Min Value
	Avg Disk Write Queue length	Transaction Log File Disk	Average Max Value Min Value
	Avg Disk Write Queue length	Database File Disk	Average Max Value Min Value
Processor	% Processor time	Database Server	Average Max Value Min Value
	% Processor time	TS Servers	Average Max Value Min Value
System	Processor Queue Length	Database Server	Average Max Value Min Value
	Processor Queue Length	TS Servers	Average Max Value Min Value
	Context Switches/sec	Database Server	Average Max Value Min Value
	Context Switches/sec	TS Servers	Average Max Value Min Value
Network Interface	Output queue length	Database Server	Average Max Value Min Value
	Output queue length	TS Servers	Average Max Value Min Value

SQL Server Access Methods	Full Scans/sec	SQL Server	Average Max Value Min Value
	Page Splits/sec	SQL Server	Average Max Value Min Value
SQL Server Buffer Manager	Buffer Cache Hit Ratio	SQL Server	Average Max Value Min Value
SQL Server Databases	Log Growths	SQL Server	Average Max Value Min Value
SQL Server General Statistics	User Connections	SQL Server	Average Max Value Min Value
SQL Server Locks	Lock Requests/sec	SQL Server	Average Max Value Min Value
	Lock Wait/sec	SQL Server	Average Max Value Min Value

Clients setup checklist

Using Navision, the business logic is executed on the client side. Therefore, it is important to maintain clean client computers.

Windows 2000 Professional vs. Windows XP Professional

Customer's Clients Setting		
Operating System (Specify number of computers)	Windows 2000 Professional	computers
	Windows XP Professional	computers
	Others	computers

Client disks configuration

	Expected Files Size	Recommended Logical Drive Letter	Customer's Logical Drive Letter
Operating System	Windows 2000 Professional 400-650MB Windows XP Professional 1-1.5GB	C	
Swap File	1.5 to 2.5 times the amount of RAM	E	

Client tools to run

	Recommendation	Customer's Policy
Disk Defragmenter	Run once a month	
Disk Cleanup	Run once a month	

SQL database setup checklist

SQL Database Files (does not apply with stripped RAID strategies)

If stripping is not used by the database disk system:

- You must create several database files
- You must store the database files in separate disks
- You can group the database files in different SQL File Groups

To tune the workload between each database file, you can assign the Navision tables to separate files (via different SQL File Groups), if the tables are involved in the same Navision transaction/process. Standard Navision Sales and Purchases posting processes involve the following Navision table types. The tables quoted with * support heavy write transactions during these processes: they must be stored in different database files.

Un-posted Document Tables:

Sales Header, Sales Line, Purchase Header, Purchase Line
Document Dimension

Posted Document Tables:

Sales Invoice Header, Sales Invoice Line, Purchase Invoice Header, Purchase Invoice Line... Posted
Document Dimension

Journal Tables:

General Journal Line, Item Journal Line, Job Journal Line
Journal Line Dimension*

Ledger Entry Tables:

General Ledger Entry, Item Ledger Entry, Customer Ledger Entry, Vendor Ledger Entry...
Detailed Customer Ledger Entry, Detailed Vendor Ledger Entry
Ledger Entry Dimension*

Register Tables:

General Ledger Register, Item Register...

Other Tables:

Value Entry*

Customer's Database Files Setting		
Number of Database Files		
Number of Disks Used		
Number of SQL File Groups		
Table split	Un-posted Document Tables	SQL File Groups Name
	Posted Document Tables	SQL File Groups Name
	Journal Tables	SQL File Groups Name
	Ledger Entry Tables	SQL File Groups Name
	Register Tables	SQL File Groups Name
	Other Tables	SQL File Groups Name

SQL indexes rebuilt

You must rebuild SQL Indexes regularly for all Navision tables. You can use the following SQL Transact to rebuild indexes:

```
SET TRANSACTION ISOLATION LEVEL SERIALIZABLE
CREATE UNIQUE CLUSTERED INDEX "CompanyName$TableName$0" ON
"DatabaseName"."dbo"."CompanyName$TableName" ("Field1InPrimaryKey", "
Field2InPrimaryKey",...) WITH DROP_EXISTING
CREATE UNIQUE NONCLUSTERED INDEX "$SecondaryKey1" ON "DatabaseName"."dbo"."
CompanyName$TableName" ("Field1InSecondaryKey1", "Field2InSecondaryKey1",...) WITH
DROP_EXISTING
CREATE UNIQUE NONCLUSTERED INDEX "$SecondaryKey2" ON "DatabaseName"."dbo"."
CompanyName$TableName" ("Field1InSecondaryKey2", "Field2InSecondaryKey2",...) WITH
DROP_EXISTING
...
IF @@TRANCOUNT > 0
COMMIT TRAN
```

You must schedule the rebuilt of indexes for a table based on:

- Whether or not records can be deleted or modified in the table.
- Whether or not records are inserted sequentially in the index.
- The number of indexes (keys) in the table.
- The number of records in the table.

SQL Indexes FillFactor

You must set the FillFactor property of an index based on:

- Whether or not records can be deleted in the table.
- Whether or not records are inserted sequentially in the index.
- The number of records in the table.

Be aware that increasing the FillFactor may increase the risk of deadlocks in SQL Server.

SQL Statistics

SQL Statistics need to be updated outside working hours, unless indexes are rebuilt with the above recommended syntax.

Navision Table Type	Recom. Rebuilt Scheduled	Recom. FillFactor	Customer's Rebuilt Schedule	Customer's FillFactor
Un-posted Document Sales Header, Sales Line, Purchase Header, Purchase Line, Document Dimension	Daily	50% All indexes		
Value Entry	Daily	70% Non-Clustered Indexes (Secondary Keys)		
Ledger Entry General Ledger Entry, Item Ledger Entry, Customer Ledger Entry, Vendor Ledger Entry..., Detailed Customer Ledger Entry, Detailed Vendor Ledger Entry, Ledger Entry Dimension	Weekly	70% Non-Clustered Indexes (Secondary Keys)		
Journal General Journal Line, Item Journal Line, Job Journal Line, Journal Line Dimension	Weekly			
All Other Standard Tables	Monthly			
Customised Tables (Specify the names of the tables and the settings)				

Customisation code review checklist

C/AL Review	Customer's C/AL Code Review (Tick if done)
<u>SETCURRENTKEY function:</u> You must always call the SETCURRENTKEY function before the SETRANGE and SETFILTER functions.	
<u>SETCURRENTKEY function:</u> You must not modified by code a field included in the current key.	
<u>SETRANGE / SETFILTER function:</u> You must only filter fields included in the current key (fields included in the SETCURRENTKEY parameter). You must set the filters following the order of the fields in the current key (holes must not be left in the key).	
<u>SETRANGE / SETFILTER function:</u> You must not modify filters in a loop, so the rowset can be re-used by the database driver.	
<u>CALCFIELDS function:</u> For the sum, average and count flowfields, the CALCFIELDS function can generate a SUM and/or COUNT SQL statement on the main table. You must not specify more than 2 sum, average or count flowfields as parameters.	
<u>CALSUMS function:</u> The CALCSUMS function generates a SUM SQL statement on the SIFT table. Before calling the function, you must filter the fields following the order of the fields in the current key (holes must not be left in the key).	
<u>INSERT function:</u> You must not test the result of the INSERT function with IF, or the bulk insert mode can not be switched on by the database driver after 5 records inserted.	
<u>COUNT function:</u> You must not use the COUNT function. Use the COUNTAPPROX function instead.	
<u>UPDATE function:</u> You must not update progress windows in each iteration of a large and tight loop.	
<u>MaintainSQLIndex property:</u> You must disable this property on a table key in situations where the key must be created to allow only occasional sorting (for example, when running infrequent reports).	
<u>MaintainSIFTIndex property:</u> You must disable this property on a table key in situations where SumIndexFields must be created on the key to allow FlowField calculations, but the calculations are performed infrequently or on small sets of data.	
<u>SIFTLevelsToMaintain property:</u> You must decrease the number of SIFT levels to maintain in situations where SumIndexFields must be created on the key to allow FlowField calculations, but the calculations are never performed according to values of specific fields on the key.	
<u>SourceTablePlacement property:</u> For tables containing hundreds of thousands records, you must set the SourceTablePlacement property to First in list forms.	
<u>List Forms:</u> You must minimise the number of Flowfields in list forms (use statistics forms instead).	

C/AL Review	Customer's C/AL Code Review <i>(Tick if done)</i>
<u>Table Locking:</u> You must always insert, modify and delete records at the end of a transaction.	
<u>Table Locking – Standard Tables:</u> To avoid deadlocks, you must lock (1 – see below) standard tables in a specific order: <ol style="list-style-type: none"> 1. Journal line 2. Batch name 3. Journal template 4. Purch. Rcpt. Header 5. Purch. Rcpt. Line 6. Sales Shipment Header 7. Sales Shipment Line 8. Sales Invoice Header/Sales Cr. Memo Header/ Purch. Inv. Header/Purch. Cr. Memo Hdr. 9. Sales Invoice Line/Sales Cr. Memo Line/Purch. Inv. Line/ Purch. Cr. Memo Line 10. Purchase Line 11. Purchase Header 12. Sales Line 13. Sales Header 14. G/L Entry 15. G/L Register Refer to specific documentation to know when a Navision table is locked.	
<u>Table Locking – Customised Tables:</u> To avoid deadlocks, you must define a locking order for customised tables: <ol style="list-style-type: none"> 1. 2. 3. 4. 5. 6. 7. 8. 	

(1) Table/Record Locking implementation:

Navision Server:

- LOCKTABLE locks the table until the end of the process
- GET, FIND, NEXT, CALCSUMS do not lock the table
- INSERT, MODIFY, DELETE lock the table until the end of the process

SQL Server:

- LOCKTABLE does not lock the table
- GET locks the record until the end of the process if LOCKTABLE, INSERT, MODIFY or DELETE are called first
- FIND, NEXT locks the recordset (+ 1 record above and 1 record below) at the beginning of the loop until the end of the process if LOCKTABLE, INSERT, MODIFY or DELETE are called first
- CALCSUMS locks the SIFT records until the end of the process if LOCKTABLE is called first
- INSERT, MODIFY, DELETE lock the record until the end of the process

Deadlocks investigation with SQL Server

Deadlocks have to be investigated on-site.

1. Turn on the trace flags in SQL Server

Turn on the following trace flags in SQL Server:

- /1204: Returns the type of locks participating in the deadlock and the current command affected.
- /T3605: Forces the information returned by other trace flags to be placed in the SQL Server error log.

In SQL Server Enterprise Manager, right click on the name of the server:

- Go to Properties
- Click on the General tab
- Click on Startup Parameters
- Add /1204 and /T3605 to the parameter list
- Stop and Start SQL Server

All deadlocks will be recorded in the SQL Error Log: C:\Program Files\Microsoft SQL Server\MSSQL\LOG\ERRORLOG for a standard installation.

The file can be opened with Notepad. Deadlocks are recorded as followed:

Deadlock encountered Printing deadlock information

```
... Wait-for graph
...
... Node:1
... KEY: 11:1125579048:1 (9700bd18b978) CleanCnt:1 Mode: X Flags: 0x0
... Grant List 0::
...   Owner:0x42b97940 Mode: X           Flg:0x0 Ref:2 Life:02000000 SPID:54 ECID:0
...   SPID: 54 ECID: 0 Statement Type: SELECT Line #: 1
...   Input Buf: RPC Event: sp_prepexec;1
... Requested By:
...   ResType:LockOwner Stype:'OR' Mode: U SPID:53 ECID:0 Ec:(0x436F7368) Value:0x42b979a0 Cost:(0/C10)
...
... Node:2
... KEY: 11:1157579162:1 (a7008a8feffb) CleanCnt:1 Mode: X Flags: 0x0
... Grant List 0::
...   Owner:0x42b97b40 Mode: X           Flg:0x0 Ref:1 Life:02000000 SPID:53 ECID:0
...   SPID: 53 ECID: 0 Statement Type: SELECT Line #: 1
...   Input Buf: RPC Event: sp_prepexec;1
... Requested By:
...   ResType:LockOwner Stype:'OR' Mode: U SPID:54 ECID:0 Ec:(0x43759368) Value:0x42b978a0 Cost:(0/B4)
... Victim Resource Owner:
...   ResType:LockOwner Stype:'OR' Mode: U SPID:54 ECID:0 Ec:(0x43759368) Value:0x42b978a0 Cost:(0/B4)
```

Using Microsoft SQL Profiler, you can also log deadlocks.

2. Analyze each deadlock encountered to know exactly what Navision processes are involved

You can find out what users are involved in the deadlock. Using SQL Query Analyzer on the database, type and execute the query:

```
exec sp_who xx
```

```
exec sp_who yy
```

where xx is the SPID of the user in Node:1 of the deadlock and yy is the SPID of the user in Node:2 of the deadlock (in bold and underlined in the ERRORLOG example above).

If the Navision Application Server is involved, the Windows account used to run the Navision Application Server will be shown as one of the users. This will be the case with Navision Commerce Portal. You can then review the Application Event Viewer in the server to check for errors at the same time. You can also review the Commerce Portal Message Log in Navision.

If an end-user is involved, you can call him directly to find out what Navision process he was running.

You can find out what tables are involved in the deadlock. Using SQL Query Analyzer on the database, type and execute the query:

```
SELECT * FROM sysobjects WHERE id = xxxxxxxxxxxx
```

```
SELECT * FROM sysobjects WHERE id = yyyyyyyyyy
```

where xxxxxxxxxxxx is the SQL table id of the Navision table in Node:1 of the deadlock and yyyyyyyyyy is the SQL table id of the Navision table in Node:2 of the deadlock (in bold and double-underlined in the ERRORLOG example above).

3. Correct the Navision C/AL code and/or report the error to Microsoft

Knowing the end users and the Navision table names involved in the deadlock helps identifying the Navision C/AL code causing it. With this information, the Navision application can be corrected.

Known issues in Navision 3.70

Codeunit 12 – application posting process

- Extensive use of the CALCFIELDS functions in triggers ApplyCustLedgEntry, CustPostApplyCustLedgEntry, ApplyVendLedgEntry and VendPostApplyVendLedgEntry.
- Navision (with Microsoft SQL Server) may run slowly when lots of customer or vendor payment transactions are posted through a batch job.

Codeunit 80 / codeunit 90 and codeunit 5704 / codeunit 5705

- Table locking orders do not match between the sales / purchases posting processes and the transfer shipment / transfer receipt posting processes.
- Deadlocks may happen when sales / purchases documents and transfer documents are posted simultaneously from different clients.

Codeunit 80 / codeunit 90 and codeunit 226 / codeunit 227

- Table locking orders do not match between the sales / purchases posting processes and the customer / vendor application posting processes.
- Deadlocks may happen when sales / purchases documents and (payment) applications are posted simultaneously from different clients.

REFERENCES

Microsoft® Business Solutions–Navision® Performance Troubleshooting Guide

Microsoft® Business Solutions–Navision® C/AL™ Programming Guide

Xtra SQL Information (white paper)

NDBCS – The Database Driver: The Microsoft SQL Server for Navision Attain (white paper)

SIFT and the Microsoft SQL Server Option for Navision Attain (white paper)

Microsoft SQL Server 2000 RDBMS Performance Tuning Guide for Data Warehousing