

TroSoft AB



Tomas Rook

2011-2012

Uppgjord		Datum	Dokument nr.
Tomas Rook		2012-01-12	Nim2SQL
Dokument typ	Fastställd	Rev.	Filnamn
Manual		А	Nim2SQL.DOC

Table of Contents

1	Ν	im2S0	QL	
	1.1	Purp	oose	.3
	1.2	Insta	allation	.3
	1.3	Lice	nse	.3
	1.4	Loca	al storage / reconnection	.4
	1.5	Турі	ical configuration	.4
	1.6	Nim	bus Generic TCP format	.4
	1.7	Nim	2SQL Setup	.5
	1.	7.1	Alarm log tab	.5
	1.	7.2	Alarm log tab	.6
	1.	7.3	System log tab	.7
	1.	7.4	Debug tab	.7
	1.	7.5	Test alarm tab	.8
	1.	7.6	Tag filter tab	.9
	1.	7.7	IP filter tab	.9
	1.	7.8	Code table tab1	0
	1.8	Nim	2SQL.INI1	0
	1.	8.1	[General] section	0
	1.	8.2	[Destination_DB1DB4] sections1	11
	1.	8.3	[Source] sections1	11

2	(1	1)
5	(1	1)

Uppgjord		Datum	Dokument nr.
Tomas Rook		2012-01-12	Nim2SQL
Dokument typ	Fastställd	Rev.	Filnamn
Manual		А	Nim2SQL.DOC

1 Nim2SQL

1.1 Purpose

Nim2SQL store alarm events to either *SQL*, *MySQL* or *ODBC*. *Nim2SQL* will run as a service and is capable of storing alarm events from *Nimbus Alarm Server* or *Alarmus Server*. It may store events to four different databases simultanously, using SQL scripts entirely defined by the user, hence it is suitable for creating own web-based alarm portals.

The link between *Nim2SQL* and the alarm event source (ex *Nimbus Alarm Server* or *Alarmus Server*) is a TCP socket. *Nim2SQL* could have any number of simultanously connections.

1.2 Installation

Install *Nim2SQL* using the installation package, ex *Nim2SQL_1_0_0_3_Setup.exe*. The installation requires admin priviliges.

Nim2SQL requires .*NET framework 2.0* to be installed, it also requires at least *MySQL* connector 6.2.4 to be installed. The *MySQL* connector is also available where *Nim2SQL* was found, the package is named *mysql-connector-net-6.2.4.zip*.

Nim2SQL is actually two programs, *Nim2SQL.exe* (the server) and *Nim2SQLSetup.exe* (setup tool). The server may be run either as service or as a normal application (in a command box). *Nim2SQLSetup* communicates with the server using a TCP socket and more than one *Nim2SQLSetup* may be run at the same time (when using terminal services).

To run *Nim2SQL* as service it first has to be configured, it requires a manual step. Start a command prompt using admin priviliges (if applicable), then run *Nim2SQL* using the '*i*' parameter, ex

Nim2SQL /i

Nim2SQL should now appear in the *Service Control Manager* services list. Change login priviliges if needed for SQL access.

To deinstall Nim2SQL as service use the 'u' parameter, ex

Nim2SQL /u

If *Nim2SQL* is not running as a service, a shortcut must manually be placed in the *Startup folder*.

1.3 License

Nim2SQL will run in demo mode with full functionality for 24 hours if it is not licensed. To purchase a license, send a mail to <u>nimbus@automatisera.nu</u> with the systemid and we will provide you a registration code. The systemid will be found on the messagebox (if not running as a service) or in the system log files.

Enter the registration code to the Nim2SQL.INI file [General]RegistrationCode parameter.

4	(1	1)
•	11	±,

Uppgjord		Datum	Dokument nr.
Tomas Rook		2012-01-12	Nim2SQL
Dokument typ	Fastställd	Rev.	Filnamn
Manual		А	Nim2SQL.DOC

1.4 Local storage / reconnection

Nim2SQL will automatically buffer events to disk if SQL server(s) are down, ensuring no events are lost. *Nim2SQL* is multithreaded and will not degrade performance if database server(s) are unavailable.

Nim2SQL may both send and expect keepalive packets to ensure alarm source link(s) are up and healthy.

All TCP sockets (when Nim2SQL is socket client) will automatically be re-established.

1.5 Typical configuration

Nim2SQL is capable of acting as socket server, when connecting to ex *Nimbus Alarm Server(s)* or as a socket client when connecting to ex *Alarmus Server(s)*.



The alarm sources in this example are three *Nimbus Alarm Servers* importing alarm events from different SCADA systems.

The alarm sources may be any application supporting *Nimbus Generic TCP* format capable of acting either as socket server or socket client.

1.6 Nimbus Generic TCP format

Nimbus Generic TCP format is text based using pipe '/' (ASCII 124) as field delimiter.

Example

 $1|2012-01-28|23:56:01|TA0220'GT41|A5|2|Pressure\ Fault|Pressure\ bad|Active$

Uppgjord		Datum	Dokument nr.	
Tomas Rook		2012-01-12	Nim2SQL	
Dokument typ	Fastställd	Rev.	Filnamn	
Manual		А	Nim2SQL.DOC	

Each event is sent as a single row and ends with line feed (LF/ASCII 10). A single LF is used as keepalive packet. Timeout for keepalives and its interval is set in Nim2SQL. Keepalives could be disabled.

1	Status. $0 = Inactive$, $1 = Active$, $2 = Acknowledge$. The field determines how the event is handled internally and what color it gets in <i>Nim2SQLSetup</i> (Green, Red, Blue)
2012-01-28	Date in swedish format <i>YYYY-MM-DD</i> . If field is omitted (empty) the server will use the PC date/time.
23:56:01	Time in swedish format <i>HH:MM:SS</i> . If field is omitted (empty) the server will use the PC date/time.
TA0220'GT41	Tag field, this field corresponds to Nimbus field [t0]
A5	Area field, this field corresponds to Nimbus field [t1]
2	Category field, this field corresponds to Nimbus field [t2]
Pressure Fault	Name field, this field corresponds to Nimbus field [t3]
Pressure bad	Description field, this field corresponds to Nimbus field [t4]
Active	Status information field, this field corresponds to Nimbus field [t5]

Above fields [t0]..[t5] corresponds to the Nim2SQL.INI replacement fields {20}..{25}.

Nim2SQL actually accepts 10 extra fields following the *t5* field, these are inserted in the replacement fields {26}..{35}.

1.7 Nim2SQL Setup

If *Nim2SQL* server is started, *Nim2SQL* will automatically connect using TCP. If *Nim2SQL* is restarted *Nim2SQL* will try to reconnect.

1.7.1 Alarm log tab

Γ	File							
	Start Nim2SQL					r IP filter. Code table		
l	Stop Nim2SQL							
		Show d	ebuginformat	ion 🕨	~	Source_Alarmus		
	Save debuginformation to Debug.txt		~	Destination_DB1				
		Exit			~	Destination_DB2		
١	201	2-01-12	10:59:55	Source_Alarmus - creat	~	Destination_DB3		
	201 201	2-01-12 2-01-12	10:59:55 10:59:55	Nim2SQLSetup - listenir All sockets and thread k	~	Destination_DB4		

File – Start Nim2SQL

Starts the Nim2SQL server (if installed as service)

File – Stop Nim2SQL

Stops the Nim2SQL server (if installed as service)

Uppgjord		Datum	Dokument nr.
Tomas Rook		2012-01-12 Nim2SQL	
Dokument typ	Fastställd	Rev.	Filnamn
Manual		А	Nim2SQL.DOC

Show debuginformation - Source_Alarmus

Check to show debug information in *Debug* tab from *Alarmus* (number of menu choices depends of defined *Nim2SQL* source sections)

Show debuginformation - Destination_DB1..DB4

Check to show debug information in *Debug* tab for each destination.

Save debuginformation to Debug.txt

Saves all information in *Debug* tab to a text file, *Debug.txt*. The file will be appended.

1.7.2 Alarm log tab

6	🗟 Nim2SQL Setup - © TroSoft 2011-2012								
	File								
0	Alarm log System log Debug Test alarm Tag filter IP filter Code table								
	Date	Time	Sender IP	Applied Filter	Destinations	Simulate	Event Id	TO (Tag)	
	012-01-12	11:00:29	127.0.0.1	* Default	DB1, DB4		9120	TA0220GT82	
	012-01-12	11:00:29	127.0.0.1	* Default	DB1, DB4		9121	TA0220GT82	
	012-01-12	11:00:29	127.0.0.1	* Default	DB1, DB4	Ja	9122	TA0220GT82	
	2012-01-12	11:00:29	127.0.0.1	* Default	DB1, DB4	Ja	9123	TA0220GT82	
	012-01-12	15:16:36	127.0.0.1	* Default	DB1, DB4		9124	TA0230GT81	
	012-01-12	15:17:22	127.0.0.1	* Default	DB1, DB4		9125	TA0230GT81	
	2012-01-12	15:17:24	127.0.0.1	* Default	DB1, DB4		9126	TA0230GT81	
	<	1111						>	

Alarm log will view the last 200 alarm events (since *Nim2SQLSetup* was started). All alarm events are also stored to the *AlarmLogFiles* folder.

The tab view all info available for each event. Color of dot indicates status, *Inactive – green, Active – red, Acknowledged – blue*

Uppgjord		Datum	Dokument nr.
Tomas Rook		2012-01-12	Nim2SQL
Dokument typ	Fastställd	Rev.	Filnamn
Manual		А	Nim2SQL.DOC

1.7.3 System log tab

🔞 Nim 2SQL	Setup - ©	TroSoft 2011-2012					
File							
Alarm log Sys	stem log De	bug Test alarm Tag filter Filter Code table					
Date	Time	Event 🔼					
2012-01-12	10:59:54	Nim2SQL is starting as a normal application in console mode					
2012-01-12	10:59:54	Using INI-fil 'C:\DATA\VB\Nimbus2\Nimbus Alarm Server\Nim2SQL\Nim2SQL\bin\Debug\\.\.\Settings\Nim					
2012-01-12	10:59:54	Nim2SQL is starting					
2012-01-12	10:59:54	Registration code invalid (SystemId '75-6D-04-FB')					
2012-01-12	10:59:55	Nim2SQL will run in 24 hour demo mode					
2012-01-12	10:59:55	Source_Alarmus - created connection for server 127.0.0.1:12444					
2012-01-12	10:59:55	Nim2SQLSetup - listening for connection attemps at port 17167					
2012-01-12	10:59:55	All sockets and thread have started					
2012-01-12	11:00:05	Nim2SQLSetup - klient 127.0.0.1:4204 connected, using encoding IBM437					
2012-01-12	11:00:12	'Destination_DB1' could not update database, error: A network-related or instance-specific error occurred while					
2012-01-12	11:00:12	Could not update event id 9118 (/Message sending succeeded (Test av larm, hälsn Rook, TroSoft)), the destina					
2012-01-12	11:00:30	Event id 9120 (ÅTERGÅTT) (TA0220GT82/20/A//Frysskydd/) matched ** Default', will be sent to destinations '					
2012-01-12	11:00:30	Event id 9120 (TA0220GT82/Frysskydd) was not added, destination Destination_DB4 is set to 'None'					
2012-01-12	11:00:32	Event id 9121 (AKTIVT) (TA0220GT82/20/A//Frysskydd/) matched ** Default', will be sent to destinations 'DB1					
2012-01-12	11:00:32	Event id 9121 (TA0220GT82/Frysskydd) was not added, destination Destination_DB4 is set to 'None'					
2012-01-12	11:00:44	Event id 9122 (ÅTERGÅTT) (TA0220GT82/20/A//Frysskydd/) matched ** Default', will be sent to destinations '					
<							

System log will view the last 200 system log messages. All system log messages are also stored to the *SysLogFiles* folder.

1.7.4 Debug tab

🐔 Nim2SQL Setup - © TroSoft 2011-2012						
File						
Alarm log System log	Debug Test alarm Tag filter	IP filter Code table				
Date Time	Source/Destination	System	Dir	Data		
2012-01-12 15:16:	37 Source_Nimbus	127.0.0.1:27009	<-	AKTIVT [2012-01-12]15:16:36]TA0230GT81]A3[1]Fr		
2012-01-12 15:16:	37 Source_Nimbus	127.0.0.1:27009	->	<13><10>		
2012-01-12 15:17:	23 Source_Nimbus	127.0.0.1:27010	<-	<197>TERG<197>TT [2012-01-12]15:17:22]TA0230(
2012-01-12 15:17:	23 Source_Nimbus	127.0.0.1:27010	->	<13><10>		
2012-01-12 15:17:	25 Source_Nimbus	127.0.0.1:27011	<-	KVITTERAT/2012-01-12/15:17:24/TA0230GT81/A3/1		
2012-01-12 15:17:	25 Source_Nimbus	127.0.0.1:27011	->	<13><10>		
2012-01-12 15:21:	46 Destination_DB1	Data Source=WIN-0	SQL	UPDATE Active SET Status = '0', Date = '2011-12-0!		
2012-01-12 15:21:	53 Destination_DB1	Data Source=WIN-0	SQL	UPDATE Active SET Status = '1', Date = '2011-12-0		
2012-01-12 15:21:	53 Destination_DB1	Data Source=WIN-0	SQL	UPDATE Active SET Status = '1', Date = '2012-01-12		
2012-01-12 15:21:	54 Destination_DB1	Data Source=WIN-0	SQL	INSERT INTO Active (Status, Date, Time, T0, T1, T2		
2012-01-12 15:21:	56 Destination_DB1	Data Source=WIN-0	SQL	UPDATE Active SET Status = '0', Date = '2012-01-12		
2012-01-12 15:21:	57 Destination_DB1	Data Source=WIN-0	SQL	UPDATE Active SET Status = '1', Date = '2012-01-12		
2012-01-12 15:21:	57 Destination_DB1	Data Source=WIN-0	SQL	INSERT INTO Active (Status, Date, Time, T0, T1, T2		
2012-01-12 15:21:	58 Destination_DB1	Data Source=WIN-0	SQL	UPDATE Active SET Status = '0', Date = '2012-01-12		
2012-01-12 15:21:	58 Destination_DB1	Data Source=WIN-0	SQL	UPDATE Active SET Status = '2', Date = '2012-01-12		
<				>		

Debug view all information from selected sources and destinations. In above example both TCP data from a *Nimbus Alarm Server* and SQL commands to a database table is visible.

Debug information could be stored to file using the *File – Save debuginformation to Debug.txt* command.

Uppgjord		Datum	Dokument nr.
Tomas Rook		2012-01-12	Nim2SQL
Dokument typ	Fastställd	Rev.	Filnamn
Manual		А	Nim2SQL.DOC

1.7.5 Test alarm tab

🔞 Nim2SQL Setup - © TroSoft 2011	-2012					
File						
Alarm log System log Debug Test alarm	Tag filter II	P filter Code t	able			
_ <u>I</u> estAlarm					1	
Description	Date	Time	Sender IP	TO (Tag)		
A-alarm test	2012-01-12	15:26:43	127.0.0.1	TA0220GT82	Test Inactive Test Active Test Acked Disable DB Set actual time	
<				>	Add Remove Update	

Test alarms could easily be created. All fields including *date/time* are editable by doubleclicking desired column/field.

Select desired test alarm and click any of the test buttons. To automatically update *date/time* stamp click *Set actual time*.

The *Update* button reloads test alarms from disk (is needed if another *Nim2SQLSetup* user changed something)

If '*Disable DB*' is checked, the alarm event will pass through *Nim2SQL* but will not be inserted into any database.

Uppgjord		Datum	Dokument nr.
Tomas Rook		2012-01-12 Nim2SQL	
Dokument typ	Fastställd	Rev.	Filnamn
Manual		А	Nim2SQL.DOC

1.7.6 Tag filter tab

🚳 Nim2SQL Setup - © TroSoft 2011	-2012			
File				
Alarm log System log Debug Test alarm	Tag filter IP filter Code table			
_ Iag filter				
Description	Tagfilter (T0)	DB1	DB2	
🔽 * Default	×	Yes	-	bbA
North	0*,K*,P*	-	Yes	
				Remove
				Update
	111			

It is possible to determine where alarm events are to be stored using a tag filter. The above example sends all alarm events to DB1. Events where tags begins with O, K or P will also be sent to DB2. Wildcards are accepted, ex *TA02GT?1 is a valid filter. Multiple filters are separated using comma.

The filters may be enabled/disabled by doubleclicking the checkbox (defaults to disabled).

Doubleclick desired database to select/deselect it.

The *Update* button reloads tag filters from disk (is needed if another *Nim2SQLSetup* user changed something)

Observe that at least one filter (either tag or ip) is needed or no alarm events will be sent to destinations databases.

1.7.7 IP filter tab

🔞 Nim2SQL Setup - © TroSoft 201	1-2012			
File				
Alarm log System log Debug Test alarn	Tag filter IP filter Code table			
_ <u>I</u> P filter				
Description	IP address filter	DB1	DB2	
Nimbus from INU	192.168.123.80	Yes	•	Add
				Hemove
				Update
			>	

It is possible to determine where alarm events are to be stored using an *IP filter*. The above example sends alarm events from *192.168.123.80* to *DB1*. Wildcards are accepted, ex *192.168.123.** is a valid filter. Multiple filters are separated using comma.

Uppgjord		Datum	Dokument nr.
Tomas Rook		2012-01-12	Nim2SQL
Dokument typ	Fastställd	Rev.	Filnamn
Manual		А	Nim2SQL.DOC

The filters may be enabled/disabled by doubleclicking the checkbox (defaults to disabled).

Doubleclick desired database to select/deselect it.

The *Update* button reloads ip filters from disk (is needed if another *Nim2SQLSetup* user changed something)

1.7.8 Code table tab

🔞 Nim2SQL Setup - ©	TroSoft 2011-2012		
File			
Alarm log System log Deb	oug Test alarm Tag filter IP filter Code ta	ble	
Code table		-	
Description	IP address filter	Table	
⊘ test	× × × × • • •	ISO-8859-1	Add Remove Update

If a specific code table is to be used, ex if you have made an own application acting as alarm source, you could change code table. It is related to the alarm source ip address.

The code tables may be enabled/disabled by doubleclicking the checkbox (defaults to disabled).

Doubleclick the 'table' column to change code table.

The *Update* button reloads code tables from disk (is needed if any other *Nim2SQLSetup* user changed something)

Code table may also be set directly in *Nim2SQL.INI* source sections and will then be the defalt code table for that alarm source.

1.8 Nim2SQL.INI

Most settings are entered in the .\Settings\Nim2SQL.INI file.

The server does not need to be restarted for changes in *destination* and *source* sections to be applied, just save *Nim2SQL.INI* file and the server will automatically use the new settings within a couple of seconds even for buffered events (ex if previous connection string was bad).

1.8.1 [General] section

This section contains some general settings, ex number of lines in each log window and registrationcode for licensing

Uppgjord		Datum	Dokument nr.
Tomas Rook		2012-01-12 Nim2SQL	
Dokument typ	Fastställd	Rev.	Filnamn
Manual		А	Nim2SQL.DOC

1.8.2 [Destination_DB1..DB4] sections

These sections contains info about the destination databases. The sections must be named *[Destination_DB1]..[Destination_DB4].*

Set parameter *DestinationType=none* if the database should be disabled.

Default commandstrings are *CommandString1* and *CommandString2*. *Nim2SQL* will first try *CommandString1* and if it fails, it will try *CommandString2*, ex

CommandString1=UPDATE Active SET Status = '{0}', Date = '{2:yyyy-MM-dd}', Time = '{2:HH:mm:ss}' WHERE TO LIKE '{20}%';

CommandString2=INSERT INTO Active (Status, Date, Time, T0, T1, T2, T3, T4) Values('{0}', '{2:yyyy-MM-dd}', '{2:HH:mm:ss}', '{20}', '{21}', '{22}', '{23}', '{35}');

In above example, *Nim2SQL* first try to update a post in table *Active*, if it fails (post is missing) it will insert it into table *Active*.

CommandStrings may be different for each status change, ex *CommandStringActive1* for active alarm events.

The replacement characters, ex $\{0\}$ are explained in the ini-file.

See Nim2SQL.INI for further instructions.

1.8.3 [Source] sections

These sections contains info about the alarm sources. The sections must be named [*Source_xxx*], where *xxx* may be any characters, ex [*Source_Nimbus*].

Alarm sources could be either TCP client sockets or server sockets. Any number of alarms sources are allowed.

Protocol=TCPAlarmReceiver creates a server socket where applications may connect, ex *Nimbus Alarm Server* using receiver type *TCP Alarm Receiver*. The server socket is multithreaded and will allow any number of simultanously applications to connect.

Protocol= *GenericTCP* creates a client socket which will try to connect to the defined ip address (or DNS name) server application. The server application could be ex *NimOPC*.