



**Server**

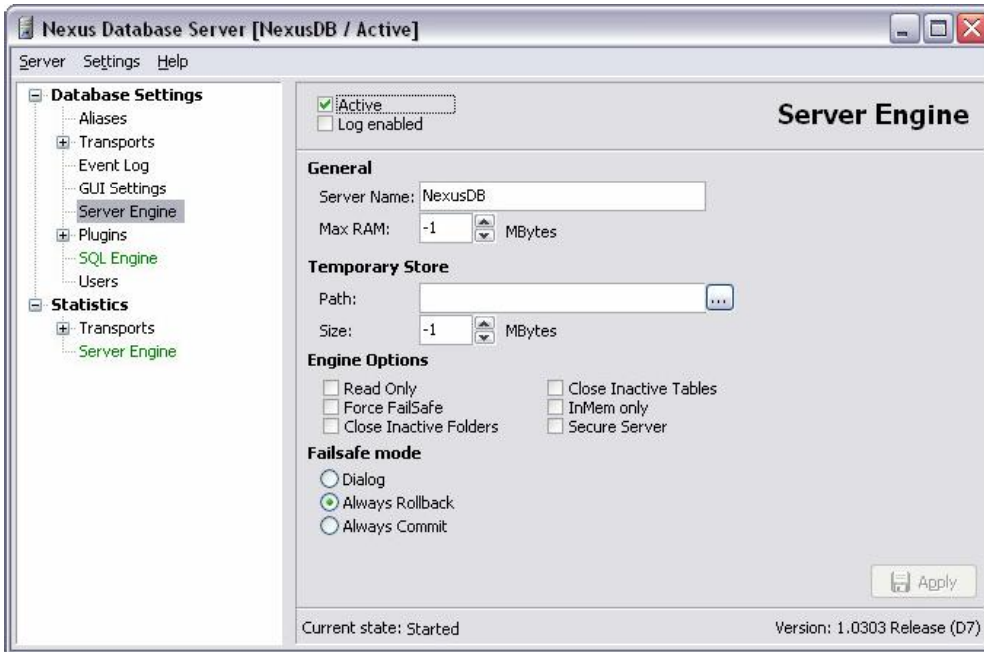
---

## **E.1 Introduction**

In this part of the manual we introduce the NexusDB Server application. This is the utility to be deployed with multi-user C/S applications. The client applications must connect to a running instance of this utility to access tables remotely.

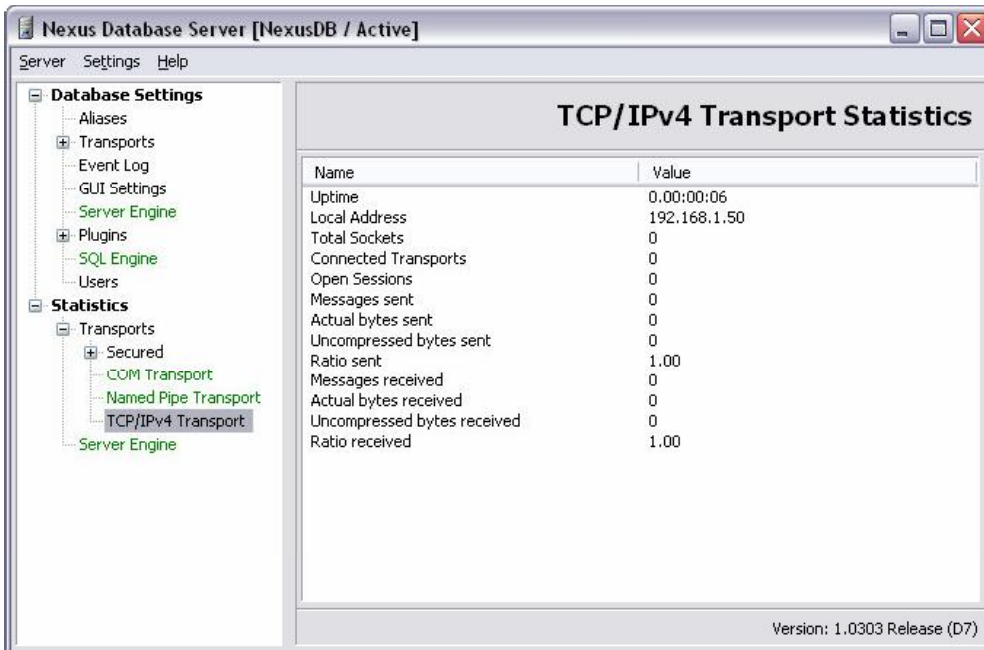
## E.2 The Server user interface

The NexusDB Server has a very clear and simple user interface. On the left hand side you find a tree view showing you all server modules that are available on this particular server. There are two main sections: Database Settings, which allows you to change the settings of each module,



**Screenshot 39: Server User Interface (Settings)**

and Statistics, which gives you an overview on the workload/status of the different modules.



**Screenshot 40: Server User Interface (Statistics)**

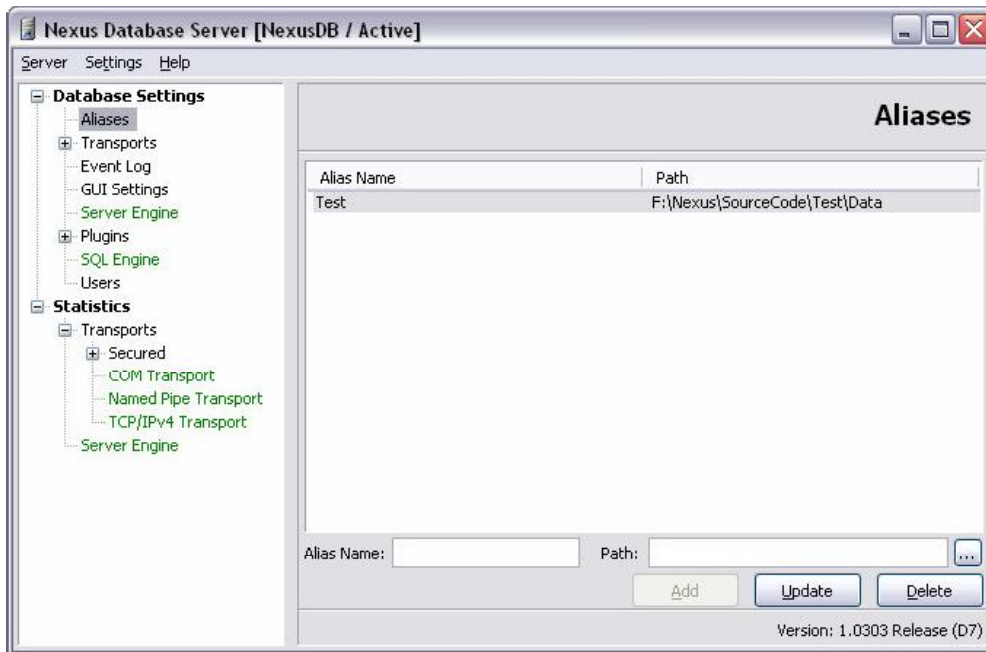
The items in the tree view are colour coded: all activated modules are shown in green in the tree view; all stopped ones orange and inactivate ones red. Black modules are stateless modules that can be changed at any time.

For a better overview and easier reading, the tree view also allows categorizing of items, as you can see in the above example on the Transports node. Click on the usual +/- symbols to expand or collapse a category level in the tree view.

## E.3 Settings Panel

In this section you will get a detailed description for all supported settings in the default Server. Please note that the server user interface is easily and almost generically extensible, thus the listing will not be complete for changed servers.

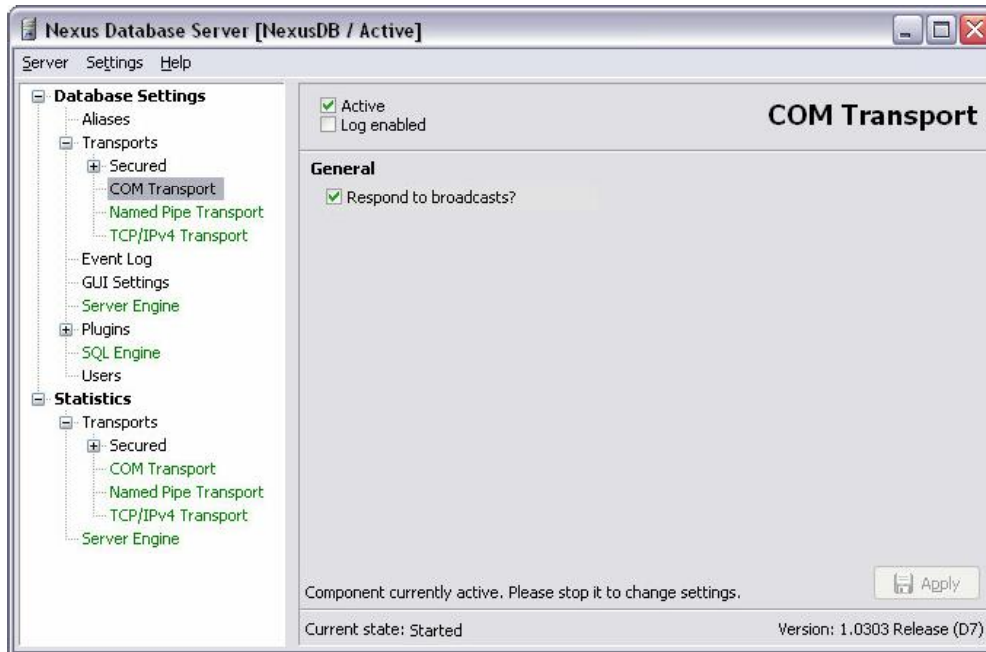
### E.3.1 Aliases



Screenshot 41: Server Settings – Aliases

The Aliases window is divided into two parts: the list of current active aliases on top and the entry fields and action buttons for adding, updating and deleting aliases below. Alias names are restricted to alphanumeric characters only. The path can either be a fixed path like “c:\testdb\” or a fully qualified UNC name like “\ourserver\nexusdatabase\testdb”. The action buttons are only activated if the according action is possible, otherwise they are disabled. Pressing the Add button creates a new alias and is **immediately** available for all connected sessions. The Update action changes Alias Name and Path or both of the selected items in the list. The Delete button deletes the currently selected item after confirmation.

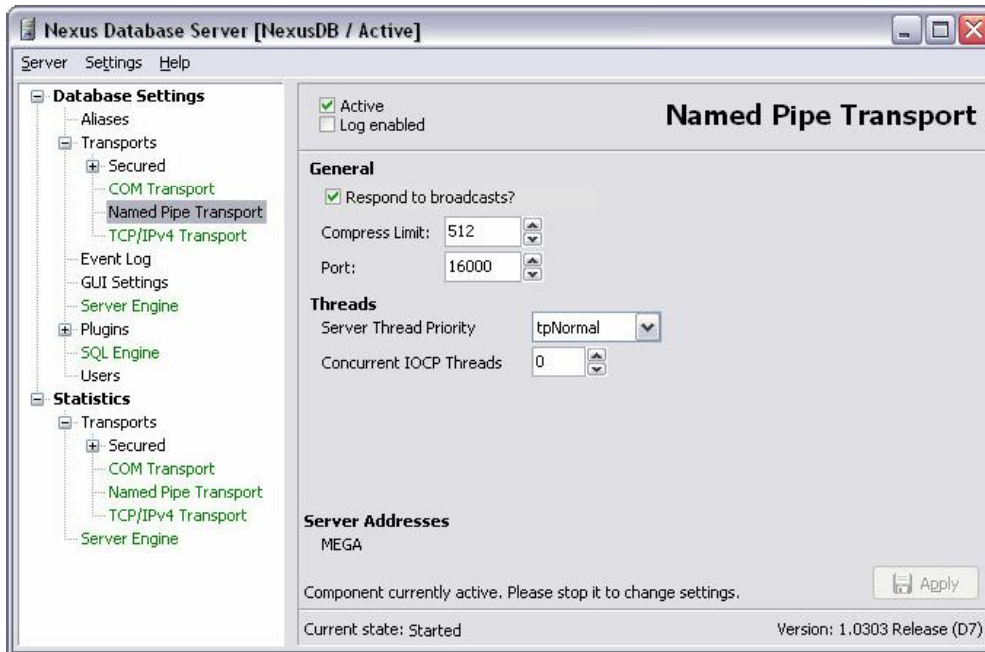
### E.3.2 COM Transport



**Screenshot 42: Server Settings – COM Transport**

The COM Transport only supports a single option, Respond to Broadcast. If you activate that option, the server sends back its identification and version number on broadcast requests received via the COM Transport.

### E.3.3 Named Pipe Transport



**Screenshot 43: Server Settings – Named Pipe Transport**

The Named Pipe Transport has some more options. Respond to Broadcast is the same as for the COM Transport. The next option, Compress Limit, sets the minimum number of bytes where the server initiates data compression for sent messages. If a message size is below that value an activated compression is ignored. **Please note** that the server always responds to compressed requests by compressing the answer, even if it is below the compression limit.

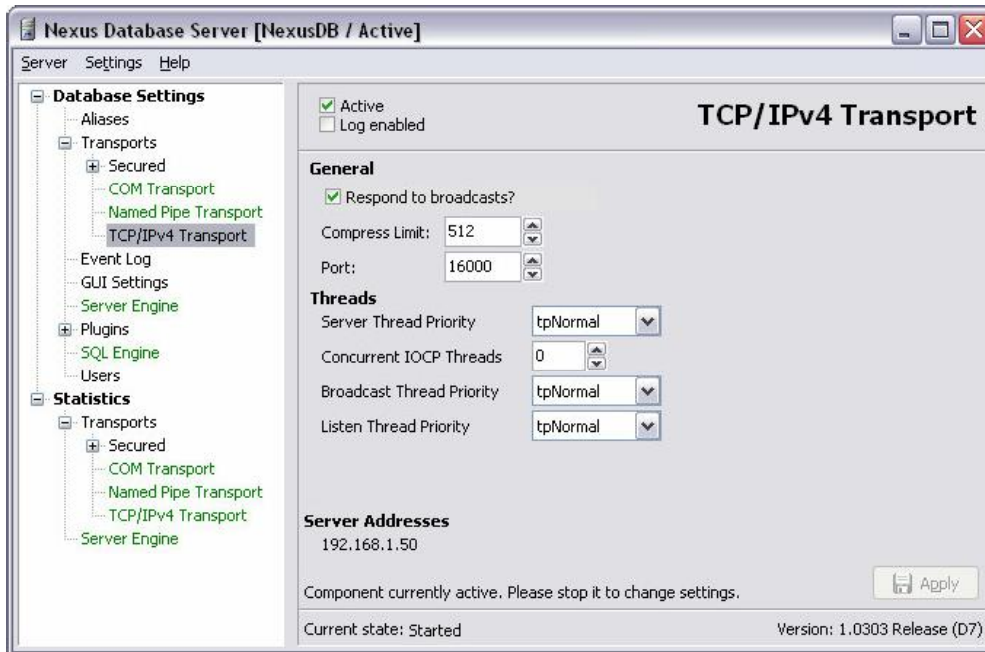
Port in Named Pipe Transport is merely a unique identifier added to the pipe name. This way it is possible to run multiple Named Pipe Protocols in one or more servers on the same machine. **Please note** that this has to match with the client side for successful communication.

The server Thread Priority sets, as the name says the priority of the server thread. The server thread of a transport is the one that actually does the work for each request. Be careful with changing thread priorities as there is a fine balance between the different threads of the server needed to achieve optimal performance. For normal processing mode you should generally leave the priorities at their default settings. Valid settings are tpIdle, tpLowest, tpLower, tpNormal, tpHigher, tpHighest and tpTimeCritical.

Concurrent IOCP Threads is the number of threads processing IOCP (I/O Completion Port) signals. If this number is 0, it means that there is one thread initialised for each processor in the server machine. For general use, this is the optimal setting. For special purposes, it might be useful to change these settings. **Please note** that this setting is ignored for machines with a Windows version below WinNT. On these machines, IOCP is not available and is emulated and ignores this setting.

Finally in the Server Addresses field you see the valid addresses of the server for this transport, which you have to use in the form ServerName@Address for connecting to this server.

### E.3.4 TPC/IPv4 Transport



**Screenshot 44: Server Settings – TCP/IPv4 Transport**

Please read about Respond to Broadcast, Server Thread Priority, Concurrent IOCP Threads and Compress Limit in the descriptions above.

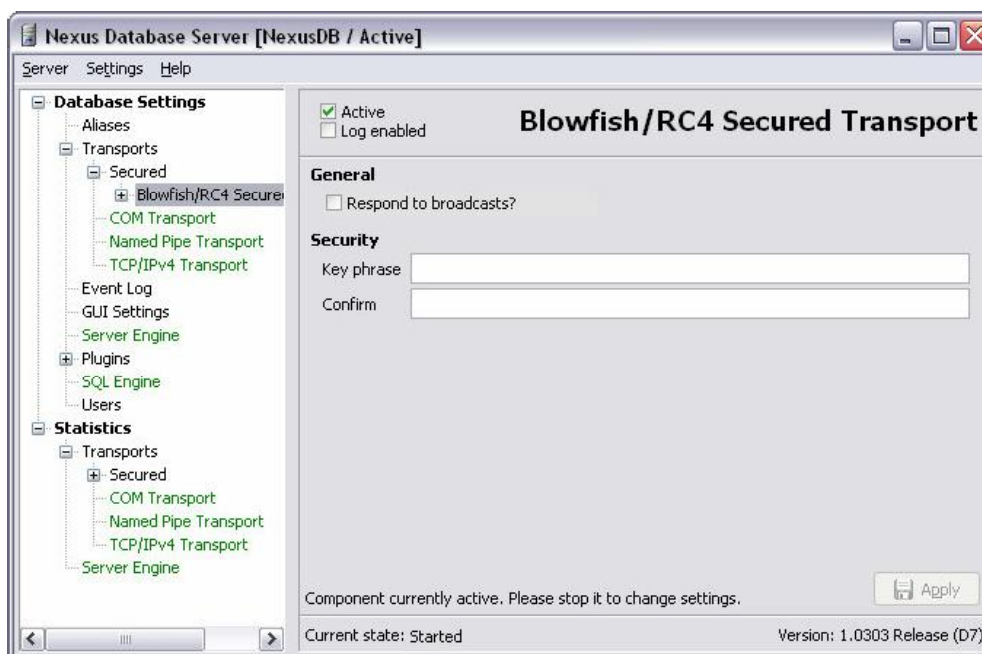
Port for TCP/IPv4 is the IP port used for communication. **Please note** that this has to match with the client side for successful communication.

Broadcast Thread Priority sets priority for the thread that is responsible for receiving and answering broadcast over for servers, while Listen Thread Priority is the one for the thread that is responsible for receiving the request and putting it into the Server Thread Queue. As mentioned before the various priorities should be left alone unless the user knows exactly what he is doing and its impact on the server performance.

Again, in the Server Addresses field you see the valid addresses of the server for this transport, which you have to use in the form ServerName@Address for connecting to this server.



### E.3.5 Blowfish/RC4 Secured Transport

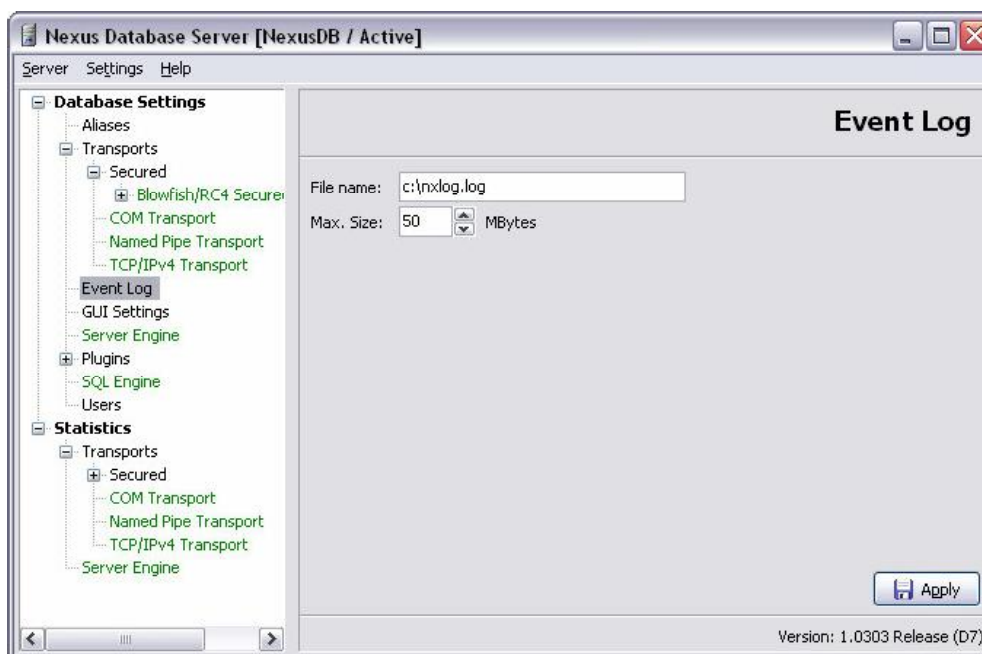


**Screenshot 45: Server Settings – Blowfish/RC Secured Transport**

The Blowfish/RC4 Secured Transport is a so-called Wrapper Transport. It encrypts requests and responses before they are sent via another regular transport like TCP or Named Pipes.

The only setting available is Key Phrase, which is used for the encryption. **Please note** that the same key phrase is needed on the client side for the communication to be successful.

### E.3.6 Event Log



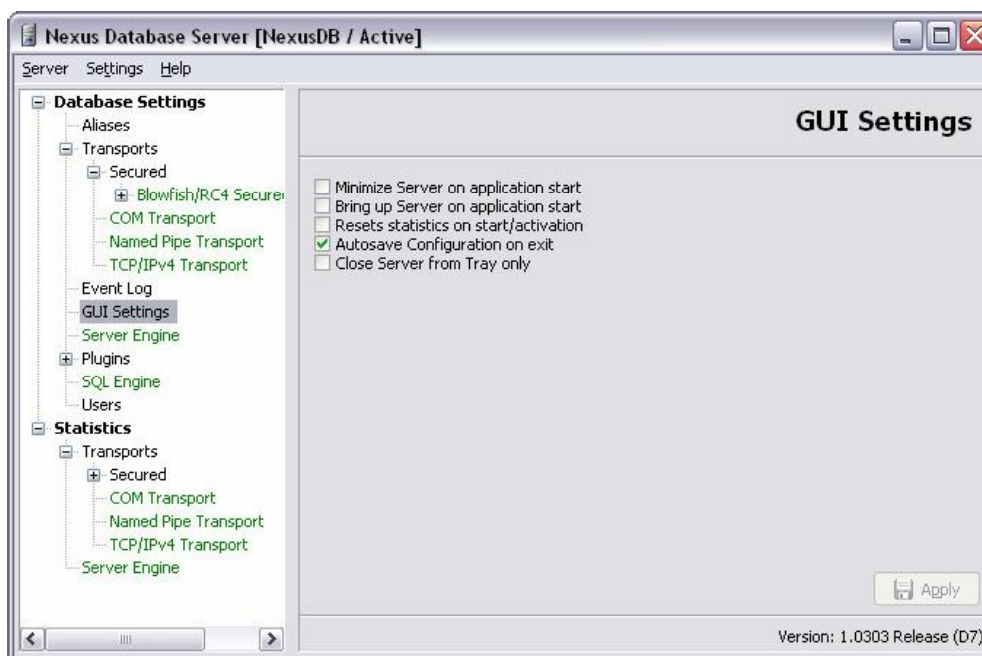
**Screenshot 46: Server Settings – Event Log**

The Event Log has two options. First one is the file name the event log is saved to and the second is the maximum size of the event log in Megabytes. The current implementation of NexusDB only writes sparse information to the log files, mainly extended errors and exception information.

All components used in the default NexusDB server are linked to the Event Log, thus all logging will end up in the given file. If the file name is empty (“”) the logging will be disabled even if the component is set to active.

If the file size reaches the maximum size it will truncate the file by 1 kilobyte at a time.

### E.3.7 GUI Settings

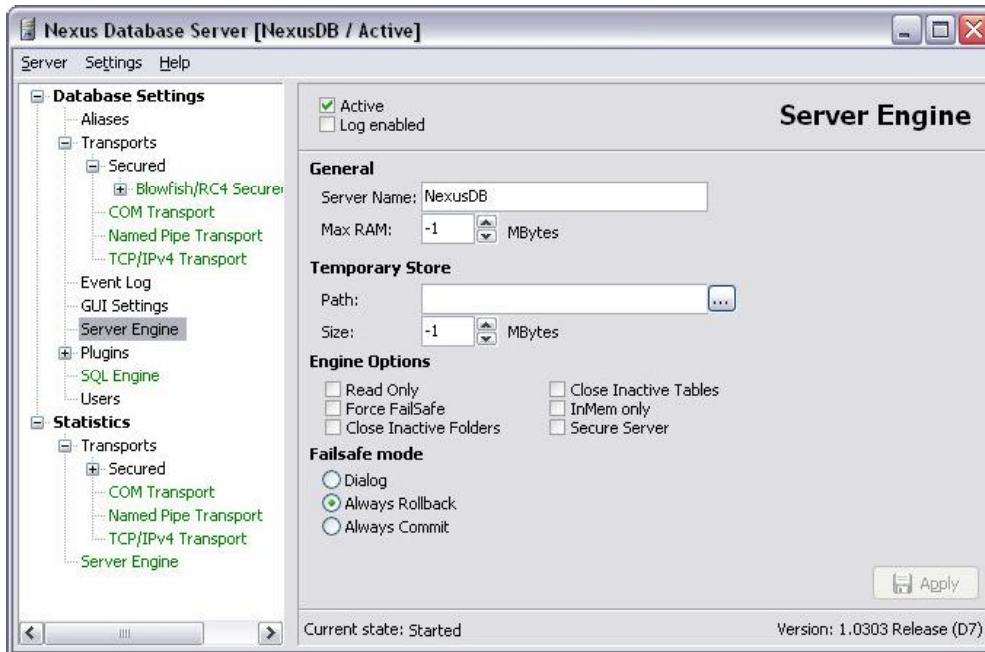


**Screenshot 47: Server Settings – GUI Settings**

The GUI settings are purely settings for the user interface and start up mode and do not influence a running NexusDB server.

The options are self-explanatory and mentioned here for completeness only.

### E.3.8 Server Engine



**Screenshot 48: Server Settings – Server Engine**

The Server Engine settings define how the core server engine operates. As a unique identifier, the server needs a name, which is entered in the Server Name option.

Max RAM defines the maximum amount of physical memory the server engine is allowed to allocate for its operation. This setting is in Megabytes and should be set. Best performance is achieved if it is set to a value large enough so that all tables fit into this memory. If this value is set to -1, the server will use a maximum of 50% of the physical memory available to the machine. This setting is very important for optimal performance of NexusDB and should be adjusted from time to time as the database size grows above its initial setting.

The settings for Temporary Storage are Path and Size. NexusDB uses its own temporary storage management and files instead of using the normal windows page file. This increases its performance and allows for dedicated virtual memory allocation for the database server. The files that NexusDB creates are stored in the directory identified by the path setting and have the maximum size stated in the size setting. The files are created as maximum size when the server engine is started with a special file system flag which makes sure they are deleted as soon as the process ends, even if it is killed (e.g. via the task manager or Delphi).

The Engine Options enable the user to change certain behaviour of the database server. The read only setting sets **all** databases and thus tables of the server to read only. This means also that the SQL data manipulation commands (insert, update, etc.) are not enabled anymore.

It is still possible though to create in-memory tables. This leads us directly to the InMem only option, which is restricting the database server to in-memory tables only. No tables are saved to disk if this option is enabled and thus if the server is closed all data is lost.

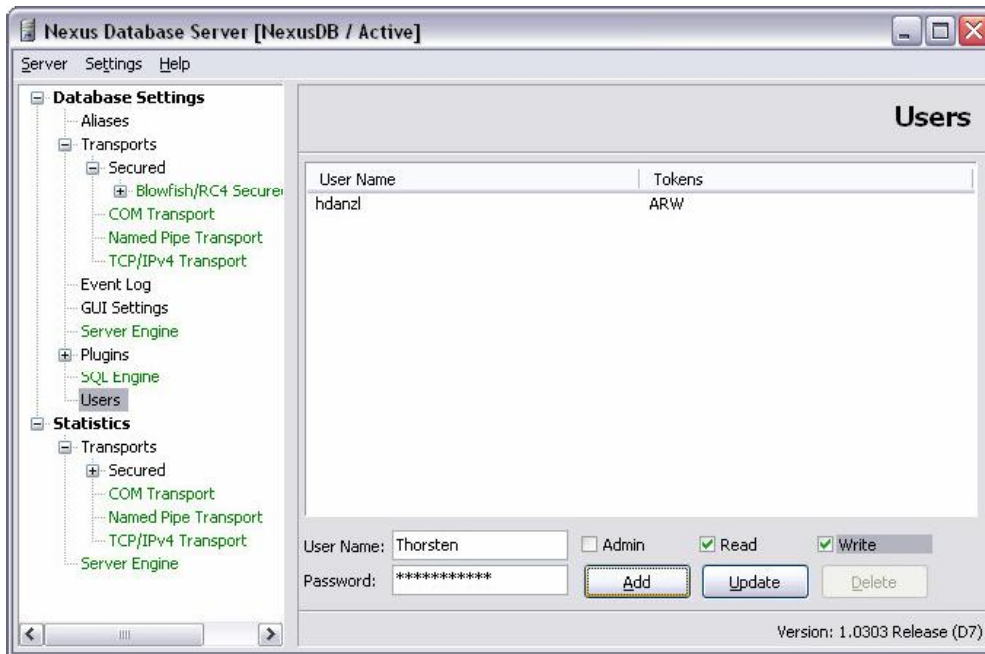
Close Inactive Tables and Close Inactive Folders influence the file caching of the server. Every open database and table in NexusDB has an exclusive file lock. This ensures that no other server or application can access the files while they are opened by the NexusDB server. In default

configuration, these two options are disabled and the server keeps the locks once they are established and thus can safely keep them in cache if there is enough room. If a client reopens the tables/databases again, the server engine can access them from the much faster memory instead of having to access them on disk again.

Force Failsafe makes the server work in failsafe mode only. Please note that this slows down the server significantly, but in certain circumstances and environment, data safety and consistency is essential. If the option is enabled the server engine writes a journal for every transaction and only if the transaction is successfully completed and the journal committed the actual transaction is considered successful. Otherwise the engine automatically applies the action set in the Failsafe mode setting. The default option is Always Rollback, which always undoes unsuccessful transactions. Always Commit is the opposite while Dialog pops up a dialog and lets the user decide what to do. **Please note** that Dialog should **NEVER** be used if the server is running as a service.

Finally the “Secure Server” option forces users to logon when establishing a connection **and** an admin user to log on to the server user interface. Please look at the following paragraph for more information on Users.

### E.3.9 Users



**Screenshot 49: Server Settings – Users**

The user options work in combination with the Secure Server setting of the Server Engine. If this option is set and at least one user with administration rights is defined, the server enforces a log on when a session or the server user interface is connected.

Adding a user is straightforward, just enter User Name, Password and select the rights for the user. Then press the Add button to create a new user. If you press update the currently selected user in the list is replaced with the settings just entered. If you click Delete, the selected line and thus the user is deleted. **Please note** that these settings are applied immediately and are not undoable.

## **E.4 Configuring and testing NexusDB Server**

Here are the steps to quickly configure the NexusDB Server and verify that it works correctly:

18. Start up NexusDB server executable (nxServer.exe)
19. Select the Server Engine item from the Database settings on the left hand tree view. The Server Engine configuration displays.
20. Make sure that a Server Name is specified and MaxRAM is set to a sensible size. You might also want to change other settings.
21. Select the Aliases item from the NexusDB Server main menu. The Alias Configuration window displays.
22. Add aliases and verify that your alias paths are defined correctly.
23. Now activate the Server modules by selecting "Start all modules" from the Server menu. You can also (de) activate single modules by selecting the item in the tree view and checking the active checkbox on the top of each configuration.
24. Make sure you have at least one transport active! (Note: The Named Pipe transport needs the server to be running on WinNT or higher. Clients are working on Win 9x/Me.)

The second step is to verify you can connect to the NexusDB Server using Enterprise Manager on the same machine. Start NexusDB Enterprise Manager and look out for the servers found. If your server name appears your server is set up correctly. This will verify you can at least see the NexusDB Server from the same machine.

The third step is to verify you can connect to the NexusDB Server from a different workstation. When Enterprise Manager starts, it sends out a broadcast for available servers. Both the TCP/IP and Named Pipes transports should show the NexusDB Server on the other machine.

## E.5 Running the Server as a WinNT service

The server can be installed as a WinNT service. For this you need to be logged in as administrator and type

```
nxServer /install
```

Or (to uninstall)

```
nxServer /uninstall
```

When installing the server it will prompt you for a user account and password the server should run as. This is expected in the format DomainName\AccountName. DomainName can be "." for the local domain. Please note the selected account must have the "logon as service" rights. You can assign these to a user account with the Microsoft® policy editor. To do this in WinXP, go to the Start menu then select Control Panel/Administrative Tools/Local Security Policy. The policy editor will start. In the tree view select Local Policies/User Rights Assignment. On the right hand panel look for "Logon As Service" and double click it. Press "Add User or Group" and add the wanted AccountName to the list of allowed accounts.

The service will be installed as Automatic; this means it will be started on any subsequent machine boot. Once the service is installed the application prompts if the service should be started immediately.

We've also added some optional parameters for your convenience. The complete set is:

Help, ?	Shows Help
Install	Installs the service
Uninstall	Uninstalls the service
Start	Start Service after installation
Nostart	Do not ask for starting service
Silent	No Dialogs at all
Username:name	User account in "domain\username" form for the service
Password:pw	password of the service account

If neither nostart nor start and silent are specified, the installation asks if the service should be started.

### Examples:

```
nxServer.exe /install
```

Will prompt for account, then install the service and ask if it should be started

```
nxServer.exe /install /username:.\test /password:test /silent
```

Will (try to) install the service with the given account. If successful it will NOT start the service or prompt for it.

```
nxServer.exe /install /username:.\test /password:test /start
```

Will (try to) install the service with the given account. If successful it will start the service and tell you that it was installed and started.

```
nxServer.exe /install /username:.\test /password:test /nostart
```

Will (try to) install the service with the given account. Will not start or prompt for starting the service and tell you that it was installed.

```
nxServer.exe /install /username:.\test /password:test /start /silent
```

Will (try to) install the service with the given account. If successful it will start the service. No messages will appear.



## E.6 Initialise the Server from the nxServer.init file

When deploying the Server to clients, NexusDB customers often need to setup Aliases, Users and default settings for the server. This is best done by integrating this into the application setup procedure. The NexusDB supports this feature by being able to read an initialisation file at start up, if it can't find a valid work settings file.

To make it possible that the server only reads from this initialisation file and never loads or saves work settings, the TnxBaseServer component has an option AutoSaveConfig. If this option is false the server does **not** save the set options, thus always uses the init file. This property is set to true in the default server, so if you want to deactivate it you will have to recompile the server.

The initialisation file is called nxServer.init and has to be created in or copied into the same directory as the nxServer executable. The file work just like a normal windows ini file, thus it's easily readable and also changeable with most setup creation tools.

Each section holds the settings for the according component or sub-engine NexusDB. Most of these settings comply with the property names of the Delphi components and their possible values can be found in the Reference section of the manual.

### E.6.1 General Section

The General section currently has two supported options:

DeleteUsersAfterRead	If equal to 1 the User section of the file is deleted after the users have been added. This makes sure that the file doesn't hold usernames or passwords longer than absolute necessary.
DeleteBFKeyAfterRead	This is the equivalent option for the BlowFish key phrase.

### E.6.2 GUI Settings

These settings define the appearance and behaviour of the user interface. The options are:

MinimizeOnStart	Set to true if the server should automatically be minimized to the tray on start up
BringUpServerOnStart	If true <b>all</b> modules that are marked as start up are activated when the server is started.
ResetsStatsOnActivate	If this option is true, every time the sever engine is activated, <b>all</b> statistics values are reset to their initial values
AutoSaveConfig	Set this option to true if you want to autosave the options every time the server is closed.
HiddenUIPages	Here you can specify a comma-delimited list of module captions. The according modules will <b>NOT</b> be visible and editable in the UI.

### E.6.3 Server Engine Section

These are the supported options for the Server Engine. For more information also look at the settings description earlier in this chapter.



ServerName	This is the unique name of server and all clients need this to connect.
MaxRAM	Maximum RAM the server uses in mega bytes
TempStorePath	The path for the NexusDB page file.
TempStoreSize	The size of the temporary storage.
ActiveAtRuntime	If this is True the Server Engine is automatically started after the initialisation
Options	Options for the Server Engine. Valid are seoReadOnly, seoForceFailSafe, seoCloseInactiveFolders, seoCloseInactiveTables, seoInMemOnly, seoIsSecure

#### E.6.4 TCP/IPv4 Transport and TCP/IPv4 Transport (Blowfish) sections

Please see TCP/IPv4 Transport Settings earlier in this chapter for detailed information on the various options.

Port	TCP port of the Transport
RespondToBroadCasts	Set to True if the transport should answer to broadcasts
BroadCastThreadPriority	The priority of the thread that answers to broadcasts. Valid are: tpIdle, tpLowest, tpLower, tpNormal, tpHigher, tpHighest, tpTimeCritical
ConcurrentIOCPThreads	Maximum number of threads initialised for handling IO completion
ListenThreadPriority	The priority of the thread that listens for connections. Valid are: tpIdle, tpLowest, tpLower, tpNormal, tpHigher, tpHighest, tpTimeCritical
ServerThreadPriority	The priority of the thread that answers requests. Valid are: tpIdle, tpLowest, tpLower, tpNormal, tpHigher, tpHighest, tpTimeCritical
ActiveRuntime	If this is True the Transport is automatically started after initialisation.

#### E.6.5 Named Pipe Transport and Named Pipe Transport (Blowfish) sections

Please see Named Pipe Transport Settings earlier in this chapter for detailed information on the various options.

Port	Unique identifier for the pipe
RespondToBroadCasts	The priority of the thread that answers to broadcasts. Valid are: tpIdle, tpLowest, tpLower, tpNormal, tpHigher, tpHighest, tpTimeCritical
ConcurrentIOCPThreads	Maximum number of threads initialised for handling IO completion
ServerThreadPriority	The priority of the thread that answers requests. Valid are: tpIdle, tpLowest, tpLower, tpNormal, tpHigher, tpHighest, tpTimeCritical
ActiveRuntime	If this is True the Transport is automatically started after initialisation.

### E.6.6 COM Transport section

Please see COM Transport Settings earlier in this chapter for detailed information on the various options.

RespondToBroadCasts	The priority of the thread that answers to broadcasts. Valid are: tpIdle, tpLowest, tpLower, tpNormal, tpHigher, tpHighest, tpTimeCritical
ActiveRuntime	If this is True the Transport is automatically started after initialisation.

### E.6.7 SQL Engine section

The SQL engine has just one option: ActiveRuntime. If this is True, the SQL Engine is automatically started after initialisation.

### E.6.8 Blowfish/RC4 Secured Transport section

The Blowfish/RC4 Secured Transport section also has just one option and that is: Key. It specifies the key phrase, which is used for the encryption. Please look at the description earlier in this chapter.

### E.6.9 Aliases section

This is a simple list of aliases. The format is

Aliasname = Aliaspath

Make sure that that the Aliaspath exists, otherwise clients can't open them.

### E.6.10 User section

The users section defines the Name/Password combinations recognized by the server and the rights these users have. The formats of these lines are

Username = Password, Rights

"Rights" is a string of length 1..3 consisting of the letters (R)ead, (W)rite, and (A) dmin.

### E.6.11 Example

Here's an example with all supported settings:

```
[General]
DeleteUsersAfterRead=1
DeleteBFKeyAfterRead=1

[GUI Settings]
MinimizeOnStart=True
BringUpServerOnStart=True
ResetsStatsOnActivate=True
AutosaveConfig=True
```

```
;HiddenUIPages=Aliases,Users

[Server Engine]
ServerName=NexusDB5
MaxRAM=20
TempStorePath=
TempStoreSize=-1
Options=seoCloseInactiveFolders,seoCloseInactiveTables,seoIsSecure
ActiveRuntime=True

[TCP/IPv4 Transport]
Port=16000
RespondToBroadCasts=True
BroadCastThreadPriority=tpNormal
ConcurrentIOCPThreads=0
ListenThreadPriority=tpNormal
ServerThreadPriority=tpNormal
ActiveRuntime=False

[Named Pipe Transport]
Port=16000
RespondToBroadCasts=True
ConcurrentIOCPThreads=0
ServerThreadPriority=tpNormal
ActiveRuntime=False

[COM Transport]
RespondToBroadCasts=True
ActiveRuntime=False

[TCP/IPv4 Transport (Blowfish)]
Port=17001
RespondToBroadCasts=True
BroadCastThreadPriority=tpNormal
ConcurrentIOCPThreads=0
ListenThreadPriority=tpNormal
ServerThreadPriority=tpNormal
ActiveRuntime=True

[Named Pipe Transport (Blowfish)]
Port=17000
RespondToBroadCasts=True
ConcurrentIOCPThreads=0
ServerThreadPriority=tpNormal
ActiveRuntime=False

[SQL Engine]
ActiveRuntime=True

[Blowfish/RC4 Secured Transport]
Key=this should be a secure key

[Aliases]
Test=F:\Nexus\SourceCode\Test\Data
Test1=c:\NexusTestData

[Users]
Hannes=def,raw
Test=abc,ra
```