Client-Server Connections

1. Introduction

This document doesn't announce any new features of GSM (Windows). Instead, it merely collates, in a single document, various information already documented the GSM (Windows) Manual, the GSM V8.1 Notes and other Technical Notes that describe Client-Server connections (both the GLOBAL.EXE to GLSERVER.EXE; and GLOBAL.EXE to SPEEDBAS.EXE connections).

Important Note: This document assumes the "latest" versions of GLOBAL.EXE (V3.4A) and GLSERVER.EXE (V3.2). Other documents (see above) describe when a particular feature was introduced in GLOBAL.EXE or GLSERVER.EXE.

2. GLOBAL.EXE to GLSERVER.EXE Connections

This section describes the various connections between a Global Client (GLOBAL.EXE) and a Global Server (GLSERVER.EXE).

2.1 GLOBAL.EXE to GLSERVER.EXE RPC Connections

The first revision of the GSM (Windows) manual described the standard Global Client to Global Server connection using the Remote Procedure Call (RPC) interface. To establish an RPC connection between a Global Server and Global Client, the Global Server must be configured to "listen" on an RPC connection by establishing the following registry settings:

..\Global\Servers\x\ProtocolSequence
..\Global\Servers\x\Endpoint

where *x* is the Global Server identification letter (i.e. node-id) from "A" to "Z".

The ProtocolSequence setting specifies the protocol used by the particular Global Server to "listen" for remote procedure calls. Obviously, the Global Client (see below) must use the same protocol. For example, if the Global Server is "listening" on the TCP/IP protocol then this value should be set to "ncacn_ip_tcp". See section 2.2.3.4.1 of the GSM (Windows) manual for a complete list of allowed protocols. This value MUST agree with the value of a corresponding ValueName in the "Client section" of the registry (see below). Note that the protocol specified determines the format of the information supplied in the Endpoint value.

The Endpoint setting specifies the Endpoint value of the particular Global Server. The format of the Endpoint is dependent on the protocol used (see above). See section 2.2.3.4.2 of the GSM (Windows) manual for further details of suggested Endpoints. This value MUST agree with the value of a corresponding ValueName in the "Client section" of the registry.

The Global Client must be configured to access the Global Server via an RPC connection by establishing the following registry settings:

..\Global\Client\Servers\x\ProtocolSequence
..\Global\Client\Servers\x\NetworkAddress
..\Global\Client\Servers\x\Endpoint

where *x* is the identification letter of the target Global Server (i.e. node-id) from "A" to "Z".

The ProtocolSequence setting specifies the protocol used by the particular Global Server to "listen" for procedure calls (see above). Obviously, the Global Client must use the same protocol as the Global Server. For example, if the Global Server is "listening" on the TCP/IP protocol then this value should be set to "ncacn_ip_tcp". See section 2.2.3.4.1 of the GSM (Windows) manual for a complete list of allowed protocols. This value MUST agree with the value of a corresponding ValueName in the "Server section" of the Registry on the Server computer (see above). Note that the protocol specified determines the format of the information supplied in the NetworkAddress and Endpoint values (see below).

The NetworkAddress setting specifies the network address of the particular Global Server. The format of the network address is dependent on the protocol used (see above). For example, if the Global Server is "listening" for procedure calls via the TCP/IP protocol then the network address value must be set to the dotted decimal address, or the servername, of the computer that is running the Global Server.

The Endpoint setting specifies the Endpoint value of the particular Global Server. The format of the Endpoint is dependent on the protocol used. See section 2.2.3.4.2 of the GSM (Windows) manual for further details of suggested Endpoints. This value MUST agree with the value of a corresponding ValueName in the "Server section" of the Registry on the Server computer (see above).

See the GSM (Windows) manual for more information regarding the format of the NetworkAddress and Endpoint values.

2.1.1 The TCP/IP RPC Protocol

The most general-purpose RPC protocol is TCP/IP. For example to configure a Global Client to access a Global Server, node-id "A", on server "globalserver" on IP address 192.168.1.109 the following registry settings must be established. On the server computer:

..\Global\Server\A\ProtocolSequence=ncacn_ip_tcp
..\Global\Server\A\Endpoint=3000

On the computer that is running the Global Client:

```
        ..\Global\Client\Servers\A\ProtocolSequence=ncacn_ip_tcp
        ..\Global\Client\Servers\A\NetworkAddress=globalserver
        ..\Global\Client\Servers\A\Endpoint=3000
```

or:

```
        ..\Global\Client\Servers\A\ProtocolSequence=ncacn_ip_tcp
        ..\Global\Client\Servers\A\NetworkAddress=192.168.1.109
        ..\Global\Client\Servers\A\Endpoint=3000
```

If the Global Client and the Global Server are running on the same computer the "localhost" IP address can be used:

```
        ..\Global\Client\Servers\A\ProtocolSequence=ncacn_ip_tcp
        ..\Global\Client\Servers\A\NetworkAddress=127.0.0.1
        ..\Global\Client\Servers\A\Endpoint=3000
```

or:

```
        ..\Global\Client\Servers\A\ProtocolSequence=ncacn_ip_tcp
        ..\Global\Client\Servers\A\NetworkAddress=localhost
        ..\Global\Client\Servers\A\Endpoint=3000
```

The TCP/IP RPC protocol (i.e. ncacn_ip_tcp) is considered general-purpose because the Global Client(s) (i.e. GLOBAL.EXE) and Global Server (i.e. GLSERVER.EXE) can execute on different computers on a network. This protocol **must** be used if the Global Client is on a different computer from the Global Server on a Windows TCP/IP network.

The recommended Endpoint values are fully described in the GSM (Windows) manual.

If the Endpoint setting for a ncacn_ip_tcp Client-Server connection is removed from the registry (i.e. by deleting the "Endpoint" valuename rather than setting the value to <blank>) the recommended value (e.g. 3000 for server "A", 3001 for server "B" etc.) will be used automatically by both the Global Client and the Global Server. For example, the registry settings described at the start of this section just become:

```
        ..\Global\Server\A\ProtocolSequence=ncacn_ip_tcp
```

and:

```
        ..\Global\Client\Servers\A\ProtocolSequence=ncacn_ip_tcp
        ..\Global\Client\Servers\A\NetworkAddress=globalserver
```

2.1.2 The Local RPC Protocol

If the Global Client is running on the same computer as the Global Server the Local RPC protocol can be used instead of TCP/IP to improve the performance of the RPC connection.

To configure a Global Server to use the Local RPC protocol:

```
        ..\Global\Server\A\ProtocolSequence=ncalrpc
        ..\Global\Server\A\Endpoint=3000
```

On the computer that is running the Global Client:

```
..\Global\Client\Servers\A\ProtocolSequence=ncalrpc
..\Global\Client\Servers\A\NetworkAddress=
..\Global\Client\Servers\A\Endpoint=3000
```

The "NetworkAddress" must be <blank>.

The Local RPC protocol (i.e. ncalrpc) is not general-purpose because the Global Client(s) (i.e. GLOBAL.EXE) MUST be running on the same computer as the Global Server (i.e. GLSERVER.EXE). The TCP/IP RPC protocol (see section 2.1.1) **must** be used if the Global Client is on a different computer from the Global Server on a Windows TCP/IP network.

2.1.3 Mixed TCP/IP and Local RPC Protocols

The configurations described in sections 2.1.1 and 2.1.2 are covered in the 1st (and currently only) revision of GSM (Windows) manual. The ncan_ip_tcp protocol on the Global Server can be used for both "remote" clients and "local" clients. In this context a "remote" client is defined as a Global Client(s) that is running on a different computer from the Global Server(s); a "local" client is defined as a Global Client(s) that is running on the same computer as the Global Server(s). The faster ncalrpc protocol can only be used for "local" clients.

To take advantage of the improved performance of ncalrpc compared to ncacn_ip_tcp, on a "local" Global Client a mixed network protocol configuration must be used. The Global Server always "listens" on a special, implicit Local RPC (ncalrpc) protocol in addition to the protocol explicitly defined in the registry. This feature allows a Global Server to service requests from both a "local" Global Client (via the implicit Local RPC connection) and "remote" Global Client(s) (via the explicit RPC connection specified in the registry i.e. normally ncacn_ip_tcp). To indicate that this automatic feature has been enabled, the following message will appear on the GLSERVER.EXE messages screen:

```
Enabling additional local RPC connection, GLSERVER_x
```

This feature allows the "local" Global Client to communicate with the Global Server using the ncalrpc protocol which is always faster than the ncacn_ip_tcp protocol. The Endpoint used by the Global Server for the "implicit Local RPC" connection is always "GLSERVER_*x*" where *x* is the server node-id (e.g. GLSERVER_A for server "A" etc.). Note that if the Endpoint key is missing from the Global Client section of the registry the default Endpoint appropriate for the local Global Server will be used. For example, the ValueNames under the key for server "A" in the Client section of the registry for a "local" client can contain either:

```
..\Global\Client\Servers\A\ProtocolSequence=ncalrpc
..\Global\Client\Servers\A\NetworkAddress=
        ..\Global\Client\Servers\A\Endpoint=GLSERVER_A
```

or:

```
        ..\Global\Client\Servers\A\ProtocolSequence=ncalrpc
        ..\Global\Client\Servers\A\NetworkAddress=
```

The registry settings for the Global Server and the "remote" Global Clients that define the explicit TCP/IP protocol settings remain the same as those documented in section 2.1.1. That is, on the Global Server:

```
        ..\Global\Server\A\ProtocolSequence=ncacn_ip_tcp
        ..\Global\Server\A\Endpoint=3000
```

On the computer that is running the "remote" Global Client:

```
        ..\Global\Client\Servers\A\ProtocolSequence=ncacn_ip_tcp
        ..\Global\Client\Servers\A\NetworkAddress=globalserver
        ..\Global\Client\Servers\A\Endpoint=3000
```

2.1.4 RPC Timeouts

The timeout period for each RPC operation can be changed from the default timeout by using either of the following registry settings:

```
        ..\Global\Client\Servers\RPCTimeout          (interface to all RPC servers)
```

or

```
        ..\Global\Client\Servers\x\RPCTimeout        (interface to single RPC server)
```

The "service specific" option (i.e. the second setting), if present, overrides the "generic server" option (i.e. the first setting) for a given server.

The timeout value can be any integer value from 0 to 10:

0 Try the minimum amount of time for the network protocol being used. This value favours response time over correctness in determining whether the server is running;

1 - 4 Behaviour (proportionally) between a value of 0 and 5;

5 Try an average amount of time for the network protocol being used. This value gives correctness in determining whether a server is running and gives response time equal weight. This is the default value;

6 - 8 Behaviour (proportionally) between a value of 5 and 9;

9 Try the longest amount of time for the network protocol being used. This value favours correctness in determining whether a server is running over response time.

10 Keep trying to establish communications forever;

Important Note: The values are not in seconds. These values represent a relative amount of time on a scale of zero to 10.

2.2 GLOBAL.EXE to GLSERVER.EXE Shared Memory Connections

Even the "faster" ncalrpc RPC protocol does not always provide the required performance on some servers. To provide the ultimate in client-server performance **for "local" clients only**, a shared memory protocol, gsmlrpc, between GLSERVER.EXE and GLOBAL.EXE has been implemented.

Important Note: The original, and now familiar, name for this protocol was "gsmlrpc"**.** This name is misleading since the protocol does not involve the Microsoft Remote Procedure Call (RPC) at all. The protocol name "gsmshm" can be used (as a preferred alternative) to "gsmlrpc" (see below).

2.2.1 Shared Memory Connection (single "local" Client)

In addition to the **automatic** implicit Local RPC (ncalrpc) protocol (see section 2.1.3) the Global Server can be configured to "listen" for requests via the Shared Memory Interface. This **option** is enabled on the Global Server by the following registry option:

..\Global\Servers\x\EnableGSMSHM

which should be used in preference to the original setting:

..\Global\Servers\x\EnableGSMRPC

where *x* is the server node-id (e.g. A, B, etc.). If this option is enabled, the following message will appear on the GLSERVER.EXE messages screen:

```
Enabling gsmshm (gsmlrpc) connection for single client
```

which is displayed instead of the original:

```
Enabling gsmlrpc connection for single client
```

To configure a local Global Client to use the GSMRPC interface, which is much faster than Local RPC and very much faster than intra-computer TCP/IP RPC, the following registry option must be set:

..\Global\Client\Servers\x\ProtocolSequence=gsmshm

which is displayed instead of the original:

..\Global\Client\Servers\x\ProtocolSequence=gsmlrpc

The following registry option must be <blank>:

..\Global\Client\Servers\x\NetworkAddress

and the following registry setting must be **absent**:

..\Global\Client\Servers\x\Endpoint

Important Note: The original Server registry ValueName is “EnableGSMRPC” whereas the original Global Client registry Value for the ProtocolSequence ValueName is “gsmlrpc”. This subtle difference is deliberate. Note that the new "EnableGSMSHM" and "gsmshm" valuenames are consistent.

2.2.2 Shared Memory Connection (multiple "local" Clients)

The "gsmshm" (gsmlrpc) protocol, described in section 2.2.1, that allows a single "local" Global Client to access a Global Server may be extended to allow multiple "local" Global Clients to access a Global Server (i.e. in an SMC configuration).

Several registry settings are required to enable the new multiple-client option.

For the "local" Global Client, the following registry setting must be enabled:

..\Global\Client\EnableMultipleClientGSMSHM=On

which should be used in preference to the original:

..\Global\Client\EnableMultipleClientGSMRPC=On

For the Global server, the following registry settings must be enabled:

..\Global\Servers\EnableMultipleClientGSMSHM=On
..\Global\Servers\MultipleClientGSMSHM*NN*=*xx*

which should be used in preference to the original:

..\Global\Servers\EnableMultipleClientGSMRPC=On
..\Global\Servers\MultipleClientGSMRPC*NN*=*xx*

where *NN* (or *N*) is a unique number and *xx* is the node-id of a local client. For example, to enable "gsmrpc" for Global clients 27,28 and 29:

..\Global\Servers\MultipleClientGSMRPC1=27
..\Global\Servers\MultipleClientGSMRPC2=28
..\Global\Servers\MultipleClientGSMRPC3=29

If this option is enabled, the following message will appear on the GLSERVER.EXE messages screen:

```
    Enabling gsmshm (gsmlrpc) connection for multiple clients
or:
    Enabling gsmlrpc connection for multiple clients
and:
        Enabling gsmlrpc connection for client 1B (27)
        Enabling gsmlrpc connection for client 1C (28)
        Enabling gsmlrpc connection for client 1D (29)
```

2.2.3 Shared Memory Connection (multiple "local" Clients) - Improved interface

The somewhat gauche registry settings described in section 2.2.2 will be replaced by a much simpler option to enable the Shared Memory Connection for multiple "local" Global Clients.

2.2.4 Shared Memory Timeouts

The timeout period for each Shared Memory operation can be changed from the default value of 5 seconds by using either of the following registry settings:

```
        ..\Global\Client\GSMRPCTimeout
or:
        ..\Global\Client\GSMSHMTimeout
```

Note that this setting, which specifies the time-out period in seconds, applies to **all** the "local" Global Servers that are accessed using the "gsmshm" (gsmlrpc) protocol. Note that this timeout value **is in seconds**, unlike the RPCTimeout (see section 2.1.4)

A problem that can only occur if the value of the "GSMRPCTimeout" or "GSMSHMTimeout" setting is too low, has been fixed in GLOBAL.EXE V3.4, and later. The problem only occurs if the time-out period for an operation on a Global Server expires before that operation completes, allowing the eventual completion of that operation to interrupt a subsequent operation on a different Global Server. This time-out should never be reduced below the default value of 5 seconds and may have to be increased, to 20 or 30 seconds, if large sub-volumes are allocated on slow filing systems (e.g. FAT filing systems).

2.2.5 Shared Memory Connection without RPC

GLSERVER.EXE V3.9, and later, recognise the following registry setting:

..\Global\Servers\x\UseRPC

The default value is "On" but this setting may be set to "Off" to disable the main RPC interface that is normally established by the Global Server (see section 2.1.1 and 2.1.2). When the main RPC interface is disabled (UseRPC=Off) the Global Server just "listens" on the "additional local RPC connection" (see section 2.1.3) and the Shared memory interface (see section 2.2.1). The "additional local RPC connection" is always enabled by the Global Server. The Shared Memory interface is only enabled by the Global Server if the "EnableGSMSHM" (or "EnableGSMRPC") option is enabled. **Consequently, the "UseRPC" option should only be set to "Off" if the Global Server is running on the same server as the Global Client(s).**

2.3 GLOBAL.EXE to GLSERVER.EXE TCP/IP Connections

In addition to the explicit RPC protocol defined in the registry (see section 2.1), the **automatic** implicit Local RPC (ncalrpc) protocol (see section 2.1.3), and the optional Shared Memory interface (see section 2.2) the V3.7, or later, Global Server can be configured to "listen" for requests via "raw" TCP/IP. This **option** is enabled on the Global Server by the following registry option:

..\Global\Servers\x\EnableGSMTCPIP

where *x* is the server node-id (e.g. A, B, etc.). If this option is enabled, the following message will appear on the GLSERVER.EXE messages screen:

Enabling GSM TCP/IP protocol

If this option is chosen then the following server registry settings must be specified:

..\Global\Servers\x\Port
..\Global\Servers\x\EnableNoDelay
..\Global\Servers\x\EnableKeepAlive
..\Global\Servers\x\ListenBacklog

The "Port" setting specifies the TCP/IP port number. If this setting is absent the "Endpoint" setting is used. **Important Note:** If both RPC TCP/IP and GSM TCP/IP are being used on the same server the Port number value **MUST** be different from the Endpoint value.

The "EnableNoDelay" option must be enabled to ensure a satisfactory performance.

The "EnableKeepAlive" option is reserved for future use.

The highly specialised "ListenBacklog" option should not be used under normal circumstances.

To configure a local Global Client to use the GSM TCP/IP interface, which is slightly faster than TCP/IP RPC, the following registry option must be set:

..\Global\Client\Servers\x\ProtocolSequence=gsmtcpip

The following registry option must be set to the computer name or dotted decimal IP address of the computer running the "target" server:

..\Global\Client\Servers\x\NetworkAddress

and the following registry setting must agree with the "Port" setting of the "target" server:

..\Global\Client\Servers\x\Port

The "Port" setting specifies the TCP/IP port number. If this setting is absent the "Endpoint" setting is used. **Important Note:** If both RPC TCP/IP and GSM TCP/IP are being used on the same server the Port number value **MUST** be different from the Endpoint value.

Two further settings are supported:

..\Global\Client\Servers\x\EnableNoDelay
..\Global\Client\Servers\x\EnableKeepAlive

The "EnableNoDelay" option must be enabled to ensure a satisfactory performance.

The "EnableKeepAlive" option is reserved for future use.

Furthermore, the following two client-side registry settings **must** be enabled:

..\Global\Client\Nucleus\EnableGSMTCPIP
..\Global\Client\Nucleus\EnableGSMTCPIPFastTransmit

By default, both of these settings are enabled and should not normally be disabled.

3. GLOBAL.EXE to Speedbase Gateway Connections

This section describes the various connections between a Global Client (GLOBAL.EXE) and a Speedbase Gateway (SPEEDBAS.EXE or SPEEDSQL.EXE). Note that details of the Speedbase Gateway registry settings, other than the connection details, are outside the scope of this document.

3.1 GLOBAL.EXE to Speedbase Gateway RPC Connections

A "remote" Global Client **must** be connected to a Speedbase Gateway using the TCP/IP RPC protocol. A "local" Global Client may be connected to a Speedbase Gateway using either the TCP/IP RPC or the Local RPC protocol.

Important Note: The Speedbase Gateway does not support the dual RPC interface described in section 2.1.3 for the Global Server. Thus, if **any** Global Client accesses a Speedbase Gateway using TCP/IP RPC then **all** the Global Clients that access that Gateway must also use TCP/IP RPC.

3.1.1 Configuring the Speedbase Gateway

The RPC options for the Speedbase Gateway are configured by the following registry settings:

```
        ..\Global\Speedbase\ProtocolSequence=ncacn_ip_tcp
        ..\Global\Speedbase\Endpoint=3100
or:
        ..\Global\Speedbase\ProtocolSequence=ncalrpc
        ..\Global\Speedbase\Endpoint=3100
```

Note the use of the recommended Endpoint value of 3100.

Many other Gateway registry settings for Microsoft SQL and Pervasive SQL are outside the scope of these notes.

3.1.2 Configuring the Global Client

The RPC options for the Global Client are configured by the following registry settings:

```
        ..\Global\Client\Speedbase\Server1\ProtocolSequence=ncacn_ip_tcp
        ..\Global\Client\Speedbase\Server1\NetworkAddress=Server1
        ..\Global\Client\Speedbase\Server1\Endpoint=3100
or:
        ..\Global\Client\Speedbase\Server1\ProtocolSequence=ncalrpc
        ..\Global\Client\Speedbase\Server1\NetworkAddress=
        ..\Global\Client\Speedbase\Server1\Endpoint=3100
```

Note the use of the recommended Endpoint value of 3100.

Alternatively, the name of the target Speedbase Gateway can be specified as a registry Value instead of a free-format, variable-text registry Key. **THIS NEW FORMAT IS STRONGLY RECOMMENDED:**

```
        ..\Global\Client\Gateways\NN\GatewayServerName
        ..\Global\Client\Gateways\NN\ProtocolSequence
        ..\Global\Client\Gateways\NN\NetworkAddress
        ..\Global\Client\Gateways\NN\Endpoint
```

The notional index number *NN* (between 01 and 99) is purely descriptive and merely serves to allow multiple Gateways to be defined in the registry.

Compare the original registry format:

```
        ..\Global\Client\Speedbase\Server1\ProtocolSequence=ncacn_ip_tcp
        ..\Global\Client\Speedbase\Server1\NetworkAddress=Server1
        ..\Global\Client\Speedbase\Server1\Endpoint=3100
```

With the equivalent alternative format:

```
        ..\Global\Client\Gateways\NN\GatewayServerName=Server1
        ..\Global\Client\Gateways\NN\ProtocolSequence=ncacn_ip_tcp
```

..\Global\Client\Gateways\NN\NetworkAddress=Server1
..\Global\Client\Gateways\NN\Endpoint=3100

Both the original and alternate registry settings are recognised by GLOBAL.EXE. However, when a connection to a Speedbase Gateway is requested, the various:

..\Global\Client\Gateways\NN\GatewayServerName=*server_name*

settings are considered before the equivalent settings:

..\Global\Client\Speedbase\server_name\

Note that the new format allows the Speedbase server name to be specified as Windows Environment Variable (i.e. it is specified as a registry value rather than a registry key).

The GatewayServerName value is always converted to upper-case by the Gateway interface module within GLOBAL.EXE.

3.1.3 Dual RPC Protocols

The Speedbase Gateway does not include the ability to "listen" on more than one RPC protocol.

3.1.4 RPC Timeouts

The interface between the Global Client and the Speedbase Gateway does not allow the default RPC timeout to be modified.

3.1.5 Multiple Threaded Interface

The following registry option is reserved for future use:

..\Global\Client\Gateways\NN\EnableGatewayMultipleThreads

3.1.6 Gateway Resets

The following registry setting is available to disable the Reset operations that are sent from GLOBAL.EXE to the Speedbase Gateway:

..\Global\Client\Nucleus\EnableGatewayReset

The default value is "On" (i.e. by default Gateway resets are **enabled**).

This setting should never be disabled unless you are explicitly advised to do so.

3.2 GLOBAL.EXE to Speedbase NLM TCP/IP Connections

A Global Client can be connected to a Speedbase NLM, running on a Novell Netware server, using the "raw" TCP/IP interface.

Important Note: The "raw" TCP/IP interface is not supported by the Speedbase Gateway.

3.2.1 Configuring the Speedbase NLM

The option to configure the Speedbase NLM for TCP/IP (i.e. instead of SPX) is beyond the scope of this document (see IN274 for full details).

3.2.2 Configuring the Global Client for TCP/IP

The "raw" TCP/IP interface is enabled by the following registry settings:

..\Global\Client\Gateways\NN\ProtocolSequence=gsmtcpip
..\Global\Client\Gateways\NN\Port=<*port number of the NLM*>

Note that the "Port" setting for the "gsmtcpip" protocol replaces the "Endpoint" setting for the various RPC protocols.

The following registry settings have the same meanings for the "gsmtcpip" protocol:

..\Global\Client\Gateways\NN\GatewayServerName=<*server name of the NLM*>
..\Global\Client\Gateways\NN\NetworkAddress=<*network address of the NLM*>

The following registry settings are also available for the "gsmtcpip" protocol:

..\Global\Client\Gateways\NN\EnableNoDelay
..\Global\Client\Gateways\NN\KeepAlive

The "gsmtcpip" interface between GSM (Windows) GLOBAL.EXE and the GSM (Novell) Speedbase NLM (SPEEDBAS.NLM) also allows the familiar "Endpoint" setting to be used instead of the new "Port" setting:

..\Global\Client\Gateways\NN\Endpoint=<*port number of the NLM*>

This provides some backwards compatibility with the interface between GLOBAL.EXE and SPEEDBAS.EXE. Note that the Port setting takes precedence over the Endpoint setting if both options are included in the registry. Note also that the Endpoint is a string (REG_SZ) value while the Port is a numeric (REG_DWORD) value;

3.3 GLOBAL.EXE to Speedbase Gateway Shared Memory Connections

A Shared Memory (SHM) interface between SPEEDBAS.EXE (and SPEEDSQL.EXE) and GLOBAL.EXE is now available. The SHM interface is considerably faster than either the local RPC (ncalrpc) or TCP/IP RPC (ncacn_ip_tcp) protocols but can only be used when the Global Client(s) is running on the same server as the Speedbase Gateway.

The SHM interface is only supported by:

GLOBAL.EXE	V3.7, or later
SPEEDBAS.EXE (non-DBX)	V2.23, or later

SPEEDSQL.EXE (non-DBX)	V2.23, or later
SPEEDBAS.EXE (DBX)	V3.16, or later
SPEEDSQL.EXE (DBX)	V3.16, or later

3.3.1 Configuring the Speedbase Gateway

To enable the SHM interface option in the Gateway, the following registry setting must be enabled:

```
..\Global\Speedbase\EnableGSMSHM=On
```

Note that the "local" SHM interface is enabled **in addition** to the "remote" protocol specified by the ProtocolSequence setting (see section 3.1). Thus, a Speedbase Gateway can be configured to "listen" for requests from both "local" Global Clients using the SHM interface and "remote" Global Clients using the ncacn_ip_tcp protocol.

The following registry setting is only recognised when the SHM interface is enabled. This setting can be used to increase the number of Shared Memory interface blocks from the default value of 10. Each Shared Memory interface block is approximately 5Kb. This value should only be increased if more than 10 "local" Global Clients (in a Symmetric Multiple Client configuration) are connected to a single Speedbase Gateway:

```
..\Global\Speedbase\NumberOfGSMSHMConnections
```

The following registry setting is reserved for future use:

```
..\Global\Speedbase\SharedMemoryID
```

3.3.2 Configuring the Global Client

The SHM interface is enabled on the Global Client by setting the following registry value to "gsmshm":

```
..\Global\Client\Gateways\NN\ProtocolSequence=gsmshm
```

When the Protocol Sequence is set to "gsmshm" the following registry settings are ignored:

```
..\Global\Client\Gateways\NN\NetworkAddress
..\Global\Client\Gateways\NN\Endpoint
..\Global\Client\Gateways\NN\Port
..\Global\Client\Gateways\NN\EnableNoDelay
..\Global\Client\Gateways\NN\EnableKeepAlive
```

When the Protocol Sequence is set to "gsmshm" the following registry setting can be used to increase the timeout period in seconds, from the default of 60 seconds, that the Global Client waits for a response from the Speedbase Gateway:

..\Global\Client\Gateways\NN\GSMSHMTimeout

Important Note: This time-out period will have to be increased when creating medium-sized and large Microsoft SQL databases.

When the Protocol Sequence is set to "gsmshm" the following registry setting can be used to increase the timeout period in seconds, from the default of 600 seconds (i.e. 10 minutes), that the Global Client waits for a response from the Speedbase Gateway for "long" operations:

..\Global\Client\Gateways\NN\GSMSHMLongTimeout

Important Note-1: The "long" time-out period is only supported by GLOBAL.EXE V3.7j, and later.

Important Note-2: The "long" time-out period may have to be increased when creating medium-sized and large Microsoft SQL databases.

The following registry settings are reserved for future use:

..\Global\Client\Gateways\N\EnableGatewayMultipleThreads
..\Global\Client\Gateways\N\SharedMemoryID

4. Summary of Connections

This section summarises the various connection options. In this section only the most basis options required for a given connection (for either server "A" or the first Gateway) are listed.

4.1 GLOBAL.EXE to GLSERVER.EXE using the TCP/IP RPC Protocol

```
..\Global\Server\A\ProtocolSequence=ncacn_ip_tcp
..\Global\Server\A\Endpoint=3000

..\Global\Client\Servers\A\ProtocolSequence=ncacn_ip_tcp
..\Global\Client\Servers\A\NetworkAddress=globalserver
..\Global\Client\Servers\A\Endpoint=3000
```

4.2 GLOBAL.EXE to GLSERVER.EXE using the Local RPC Protocol

```
..\Global\Server\A\ProtocolSequence=ncalrpc
..\Global\Server\A\Endpoint=3000

..\Global\Client\Servers\A\ProtocolSequence=ncalrpc
..\Global\Client\Servers\A\NetworkAddress=                  (i.e. blank)
..\Global\Client\Servers\A\Endpoint=3000
```

4.3 GLOBAL.EXE to GLSERVER.EXE using the Shared Memory Interface

```
..\Global\Server\A\ProtocolSequence=???                            (i.e. don't care)
..\Global\Server\A\Endpoint=???                          (i.e. don't care)
..\Global\Servers\A\EnableGSMSHM

..\Global\Client\Servers\A\ProtocolSequence=gsmshm
..\Global\Client\Servers\A\NetworkAddress=
```

4.4 GLOBAL.EXE to SPEED*xxx*.EXE using the TCP/IP RPC Protocol

```
..\Global\Speedbase\ProtocolSequence=ncacn_ip_tcp
..\Global\Speedbase\Endpoint=3100

..\Global\Client\Gateways\01\GatewayServerName=Server1
..\Global\Client\Gateways\01\ProtocolSequence=ncacn_ip_tcp
..\Global\Client\Gateways\01\NetworkAddress=Server1
..\Global\Client\Gateways\01\Endpoint=3100
```

4.5 GLOBAL.EXE to SPEED*xxx*.EXE using the Local RPC Protocol

```
..\Global\Speedbase\ProtocolSequence=ncalrpc
..\Global\Speedbase\Endpoint=3100

..\Global\Client\Gateways\01\GatewayServerName=Server1
..\Global\Client\Gateways\01\ProtocolSequence=ncalrpc
```

```
..\Global\Client\Gateways\01\NetworkAddress=                (i.e. blank)
..\Global\Client\Gateways\01\Endpoint=3100
```

4.6 GLOBAL.EXE to SPEED*xxx*.EXE using the Shared Memory Interface

```
..\Global\Speedbase\ProtocolSequence=???                    (i.e. don't care)
..\Global\Speedbase\Endpoint=???                                 (i.e. don't care)
..\Global\Speedbase\EnableGSMSHM=On

..\Global\Client\Gateways\01\ProtocolSequence=gsmshm
```