

# Performance Tuning – ScriptBase Client-Server

## Why might this be needed?

The client server mode of ScriptBase is more complicated than standalone and there are more 'moving parts' which can affect performance. When first setting up a client server installation you can safely use default settings, later you might want to consider performance tuning. It is also worth noting that new releases of ScriptBase will include changes which improve performance – but the rest of this document focuses on what IT staff supporting ScriptBase implementations can do.

## Understanding the Benchmarking utility

ScriptBase now has an option on the 'Utilities' menu for people logged in with admin rights to run a benchmarking routine. This performs a series of highly intensive tasks and measures the time to complete, providing a total benchmark index. This is extremely useful for performance tuning because you can get a benchmark, make configuration changes and check whether these have improved or worsened performance.

Please note that because the benchmark interacts with the database (and data used varies) it is not *precise*. Do not fret about getting a slightly different result each time you run: changes of just 10 points or so in the index result are probably not meaningful.

Also, the 'Archive 1000' test will not work properly until you have printed over one thousand prescriptions and the 'Validate Rx' test needs at least three valid prescriptions on the system.

Before you start making changes, run the benchmark twice and make a note of the second set of results. These are automatically copied to the clipboard so you can just paste into a document as a record.

The overall index is probably the most useful part of the result, but it can be useful to understand the individual metrics:

Ping 1000 – makes 1k ping requests to `sbase_server.exe`. Measures network latency all the way from the client software to server software. On small fast LAN expect a benchmark of around 400.

Archive 1000 – performs a query of one thousand records from the printed script archive. Tests 'pure' database SQL performance.

Open Close - opens and closes all database tables several times.

Client Maths – performs some fairly demanding maths to benchmark the abilities of the CPU on the client machine.

Disk I/O – performs a large number of 'writes' to a file in the directory holding the database. This is on the local machine in standard mode and on the server in Client Server mode.

Validate Rx – takes a sample of valid scripts and runs the very intensive ScriptBase validation routines. This is a very 'real world' measure that correlates directly with the user experience of latency.

Results are in milliseconds - lower figures are better. In developing the benchmark, the figures for each section were calibrated to be *roughly* 100-150 on the developers own machine, running in Client Server mode with the server also on the local machine. The utility of the benchmark is not in the 'absolute' figure, but to compare before and after as changes are made to the program or configuration settings.

With a benchmark under your belt, you can begin making changes. For optimal results, make one change and then test again. If you don't have much time, benchmark, make the recommended changes and test again.

### **Client Settings**

Along with sbase.exe you have the localsettings.ini file – which includes ClientServer=1 to switch the software to client server mode.

The setting with the biggest impact appears to be Compression. In future this will default to 0 (off) since in most cases it seems that compression is *detrimental* to performance. In situations where network bandwidth is very constrained and is the main performance bottleneck, compression may offer benefit. For fast networks turn it off entirely (Compression=0)

Encryption also has an impact. If your network pipe is already encrypted (as it will be if you are using a VPN 'layer') then you should *always* set Encrypt=0 as encrypting a second time hits performance for no great benefit. Only use Encrypt=1 where this is necessary.

The BufferMulti setting has a more marginal impact and a value of 6 seems – on available data – to be optimal for most sites.

The RemoteHost setting accepts both IP addresses and network names. Although our testing has not found a measurable difference, there is a theoretical advantage to using IP addresses (eg. 192.168.1.434 rather than SB\_MACHINE). If it is easy to do so, we recommend using IP address and if you need to use machine name, use benchmarking to check the impact on performance.

### **Server Settings**

Along with sbase\_server.exe you have a sbase\_server.ini configuration file.

'Server Thread Cache Size' has been found to have a fairly dramatic impact on performance and the previous default of 10 was too small. Future versions of the server software will have a higher default but you should check the value is turned on (ie remove semi-colon from the beginning of the line) and set to at least 30. It is possible that even higher values will be of benefit at sites with lots of workers but this is still being investigated.

See notes above about Encryption settings. For clients to request an un-encrypted connection the server setting must be 'Encrypted Only=0' – avoid making encryption mandatory unless all your network pipes to ScriptBase are bare bones TCP/IP and thus un-encrypted.

The BufferMulti setting has a more marginal impact and a value of 6 seems – on available data – to be optimal for most sites.

## **Network infrastructure**

Beyond coding of the application itself, the biggest factor affecting CS performance is the network. Having a high bandwidth, low latency pipe will allow data to move around faster.

ScriptBase uses a standard TCP/IP connection though this can sit within another layer of abstraction, such as VPN software. In many cases, your network transport layer will encrypt all data and it is unnecessary and computationally expensive for ScriptBase to do so as well. It is important to understand whether the pipe between `sbase_server.exe` and `sbase.exe` is encrypted at the network level and if it is, turn of encryption within `localsettings.ini` and `sbase_server.ini`

## **Database location and disk**

The best scenario is that your server machine has a local SSD disk, so that `sbase_server.exe` is reading and writing to an ultra-fast disk.

If you are writing data to a network drive it is strongly recommended that you run a comparison with the alternative of having the data on the server machine. Look at the Disk I/O metric which you want to be as low as possible. A local SSD will provide a value in the region of 100 ms.

## **Updates and feedback to the developer**

Client Server performance is a high priority for subsequent releases of ScriptBase and it is expected that substantial improvements can be made to how data is requested from the server and cached locally. When you upgrade to new versions it would be very helpful if you could benchmark immediately before and after the upgrade and provide us with feedback on how the upgrade affected the index.

If you identify any particular change to your settings or architecture that had a particularly *dramatic* impact on performance this would also be of interest to the developers so that we can hone future advice.

[support@scriptbase.co.uk](mailto:support@scriptbase.co.uk)

Version: 29 October 2018