

Quick Notes:

Enhancing performance of SQL server 2005 or 2008 transactional replication

In order to improve initial snapshot delivery performance one should do the following.

Use SQL 2005 or SQL2008 Enterprise

Use the 64 bit edition

Use a separate distributor (64 bit) with as much memory as you can garner as it drives the memory into the system cache when creating a large snapshot. Set your SQL server min memory to an appropriate level so as not to bottom it out.

The database replication snapshot creates files the same size as the files in the database to be replicated and initialization is quick due to sparse file use. You may want additional disk layouts if your system is busy. By default the rss files are put in the same directories as the physical files.

After or during publication setup change the `synch_method` to database snapshot instead of concurrent. Why do this? You cannot use more than 1 thread for `maxbcpthreads` when you have a concurrent sync method regardless of what you set the `maxbcpthreads` to for the `distrib` parameter switch.

Set your `MaxBcpThreads` to an acceptable level depending on the processors. Most systems have at least 4 to 8 processors. Start at that.

Set your `SubscriptionStreams` to an acceptable level depending on the processors. Most systems have at least 4 to 8 processors. Start at that. This is the ongoing number of threads that are used during subsequent transactional replication. See BOL for more on this parameter and when to use\ not use it.

When you run the snap a database replication snapshot will appear with exactly the same characteristics and will have files equal in size to that of the database files and will be created sparse. So all told, using the database snapshot method will increase your data file size by a factor of 200%, i.e. if your database is 100GB (data files only), you will have a replication database snapshot of 100GB.

Set your `distrib agent` to use an appropriate value for `maxbcpthreads` say 16 for a 16 processor server.

Set it to use `InProcLoader` to use bulk insert instead of `bcp`. Check the viability of doing this by seeing if you have `nvarchar(max)` or similar data types. In one instance the `bcp` file worked for `bcp` but it would not work for bulk insert on a table with that datatype.

Test the trade off of using compressed vs. non compressed snapshots. The snap will take longer to gen but may decrease transfer time across a WAN with long latencies, think your remote site. The maximum size is 2GB. Break it down into multiple publications if it goes over and just be sure that RI does not cross boundaries.

Test the use of FTP if you company allows it.

REPLICATE THE Execution OF STORED PROCEDURES AND FORCE AD HOC SQL INTO STORED PROCS.
WHY???? You save bandwidth and precious time during normal replication and also during the initial snapshot. You replicate the Execution and NOT the data!!!

Think about what gets replicated before you run a snapshot.