



How do I ensure that the Exaquantum/Batch database does not fill up?

The QBatch database is different to the other Q databases and depends on the version of Exaquantum/Batch which was used to create the database initially.

Prior to R2.50.30 the database consisted of a master data file (mdf) and 2 normal data files (ndf) all in a single PRIMARY data group and the transaction log file (ldf).

In R2.50.30 and R2.50.40 the database consisted of an mdf and ndf file in the PRIMARY data group and a further ndf file in another data file called QDATA plus the transaction log file.

All Batch data goes into the PRIMARY data group regardless of the version. This means that the QBatch2.ndf file on systems that were first installed using R2.50.30 or R2.50.40 will never be used. On these systems it is recommended that the QBatch2.ndf file be removed and a new file added in the PRIMARY data group. This has been reported as a bug and will be fixed in a future release.

All data files are set to grow by 10% with no maximum files size specified.

This means that SQL Server will expand the database files whenever this is required.

This means that as long as there is physically enough disk space available the QBatch Database should never have a problem.

It is however important to monitor the disk space usage to ensure that this is not used up completely. Use of the Exaquantum PIMS Status Monitor tool can help here. This does not support the QBatch database specifically but does allow physical disk space to be monitored and alarms to be raised if this falls below a user defined limit. See Chapter 19 of the Exaquantum Engineering Guide Volume 3.

The initial size of these database files are as below:

Filegroup Name	Logical File Name	Space Reserved
PRIMARY	QBatch	10.00 MB
PRIMARY	QBatch1	5.00 MB
PRIMARY	QBatch2	5.00 MB



This is not ideal and will cause twice as many writes to QBatch file as the other two due to the proportional fill algorithm used by SQL Server. As each file also auto grows independently this can compound this situation.

Multiple database files can improve performances on systems with multiple cores available.

It is therefore recommend to set all three files to have the same size and auto grow settings. SQL Server will then round robin the writes to each file in turn.

The required size will depend on several factors including the number of Batches, number of Master Recipes, Plant Hierarchy, the amount of formula item data and the number and size of reports.

For example, setting the initial size of each data file to 20MB with an absolute 2MB growth and a 40MB maximum. This would give an initial size of 60MB and a maximum of 120MB. In most cases the setting would be much larger than this.

Using a windows task or SQL Server job it will be possible to check these to see whether they are it is a good estimate and make any adjustments required.

A database size task which can be added to the Windows Task Scheduler is available from the local Yokogawa Affiliates.

On most systems an estimate of the amount of disk space used per Batch can be obtained by divide the disk usage increase for a particular period with the number of batches run over the same period. As in most cases it is the Batch data which accounts for the increase. This assumes that

1. All reports are batch reports
2. Any changes made to Master Recipes or Equipment Hierarchy are small compared to the number of batches.

This figure can then be multiplied by the length of time that data is required online to give a good maximum size for the database.

It is not possible to rebalance data file usage.

Exaquantum /Batch Releases Affected
All