

Description: Tips for optimizing the performance of BridgeWorX and ReportWorX.

OS Requirement: Win XP/Server 2003/ Server 2008/Vista/Windows 7

General Requirement: Installation of BridgeWorX, ReportWorX, and SQL Server 2005/2008.

Introduction

The scope of this application note is to present a number of performance optimization tips for BridgeWorX and ReportWorX, which will help end users get the most out of their applications. Typically, performance can be improved in the following areas:

- Data source optimizations
- Configuration and runtime database optimizations
- Miscellaneous optimization tips

Data Source Access Optimization

Timeouts

Access to your most important data sources can be optimized by utilizing the available data source timeout settings in the BridgeWorX and/or ReportWorX Configurator. The purpose of the timeouts is to make sure that BridgeWorX and/or ReportWorX wait long enough for data to be available prior to processing reports and transactions. If the data are available prior to the timeout expiration, BridgeWorX and ReportWorX runtime engines will not wait for the full duration of the timeout period and they will continue processing the report or the transaction immediately. Typically there are OPC Data Access and database command timeouts, as well as Read and Write timeouts as applicable. The following figure shows a BridgeWorX OPC Connector and its configurable timeouts.

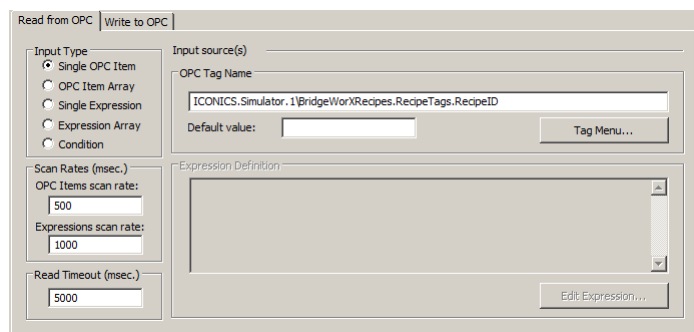


Figure 1 - BridgeWorX OPC Read Connector timeout settings

Figure 2 shows a ReportWorX configuration dialog with OPC read and database command timeouts.

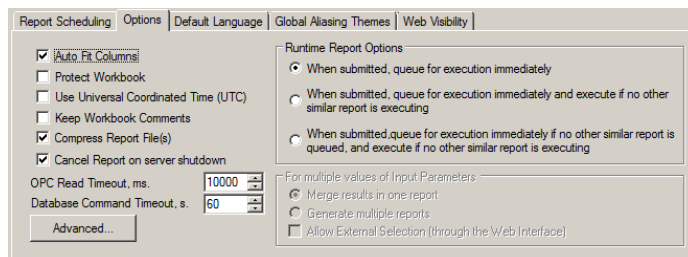


Figure 2 - ReportWorX OPC and database timeout settings

GenClient caching

As far as database data source access is concerned, both BridgeWorX and ReportWorX are utilizing the built-in caching support and connection pooling support of ADO.NET for database access, so there is not much the user needs to worry about there.

As far as OPC DA data sources, in the case of very frequent access of OPC DA tags, there is additional client-side caching that can be utilized in the ICONICS GenBroker configuration utility.

NOTE: This caching feature does NOT require GenBroker to be running.

From the Start> Programs>ICONICS Tools, the user can launch the GenBroker Configurator in order to create a new GenBroker configuration file. The user should click on the New button to activate OPC Item caching under the Advanced > Common > Optimization tree, as shown in figure 3.

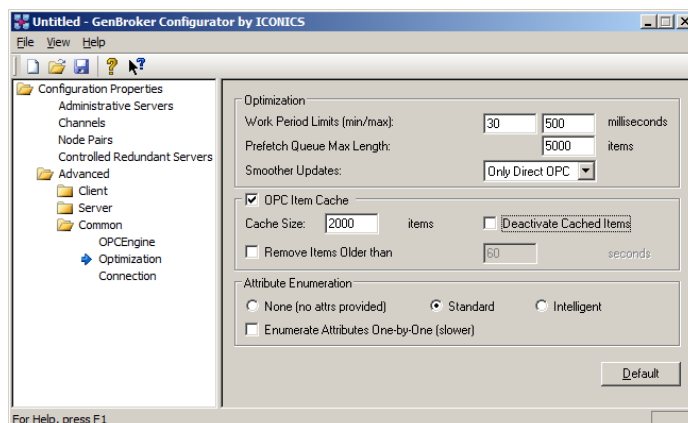


Figure 3 - Setting GenClient client side caching

To enable client-side caching, the user should check the OPC Item Cache checkbox and **uncheck** the Deactivate Cached Items options, for the cases of very frequent data access applications where the cached OPC DA tags do not process many updates between successive processing by BridgeWorX or ReportWorX.

If GenClient caching is not enabled, both BridgeWorX and ReportWorX will release the OPC DA tags upon completing each individual report or transaction. The user may need to restart BridgeWorX and ReportWorX for the client-side caching of OPC DA tags to go into effect. In addition, the user should keep in mind that this client-side caching has a global effect on all GENESIS32 applications running on the same node as BridgeWorX or ReportWorX.

In general, BridgeWorX has been tested with a moderate load of transactions triggering every few minutes on a dual core PC. If the transactions are being triggered faster than what BridgeWorX can process, it will queue them up for execution, which in return can result in considerable performance degradation. To monitor performance, ICONICS recommends using MonitorWorX to show the number of executing and queued transactions. Once the number of executing transactions starts decreasing, this may be an indication of application overload.

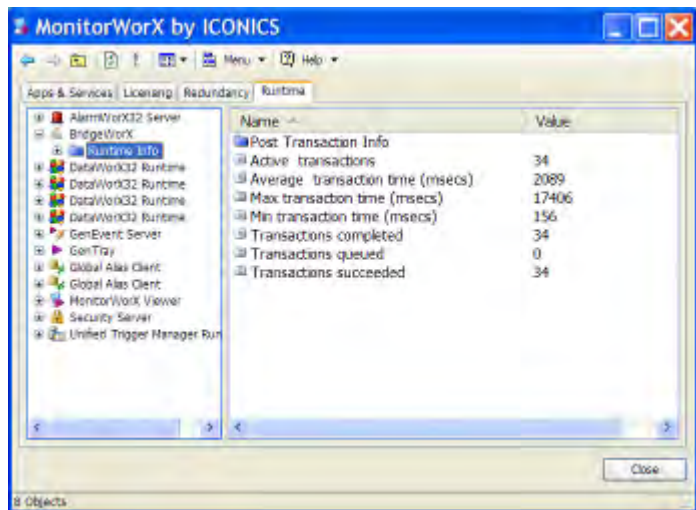


Figure 4 - Monitoring BridgeWorX through MonitorWorX

IcoWebClient caching

It is recommended to use the new OPC Request Caching feature. The caching can be configuring in the advanced properties of the OPC DA data connector in BWX or the OPC data source in RWX. The following figure shows a BridgeWorX OPC Request Caching dialog.



Figure 5 - BridgeWorX IcoWebClient caching setting

Figure 26 shows a ReportWorX OPC Request Caching dialog.

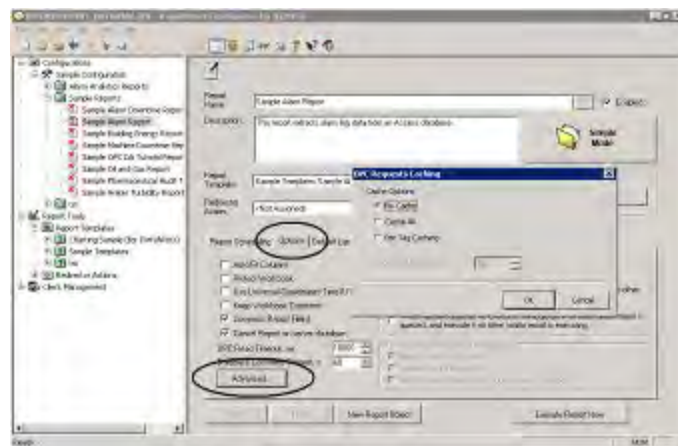


Figure 6 - ReportWorX IcoWebClient caching setting

The user can also configure a timeout interval, which is the interval for which the OPC tag will be held on the cache. Typically, the timeout interval should be much larger than the transaction frequency or report periodicity. If set 0, the cache may hold the tag for ever. Typically, using the new cache will result in about 15% performance improvement, at the expense of increased CPU utilization as well as memory utilization.

The cache is on a per thread basis – this means that if BWX/RWX are configured to use say a thread pool of 5 threads, then the cache will be duplicated across all threads (over time), which also means that the OPC server will have 5 times the number of connected clients. Since the cache is duplicated across multiple threads, it is possible that one transaction or report may be executed before the cache is updated because another transaction wrote to the OPC server. Users should implement some sort of a handshake logic (perhaps via post transaction actions) to ensure that the data are written

completely to the OPC server prior to assuming that they are already available.

Configuration and Runtime Database Optimization

BridgeWorX and ReportWorX both utilize a SQL Server 2005 or 2008 database for configuration settings and runtime state maintenance and storage. In addition, all generated reports or transaction trace files are stored internally within the ReportWorX and/or BridgeWorX configuration databases. As such, it is very important that these databases be set for optimal performance. Here are some of the recommended steps:

Set initial database file size and growth pattern

In the case of frequent report or transaction execution, especially with large reports or complicated transactions, it is recommended that the user sets the initial sizes of the BridgeWorX and ReportWorX databases to reasonably high and allows for a reasonable growth pattern as well. All of the above can be easily accomplished using the SQL Enterprise Manager. ICONICS recommends that the initial size of the BridgeWorX or ReportWorX databases should be set to something reasonable like 512 MB and it should be configured to grow by a certain value, i.e. 25 to 60 MB. Ideally, the user should *avoid* using the Grow by % option, since this can create runtime performance issues for BridgeWorX and ReportWorX application in the cases that their databases sizes are already very large.

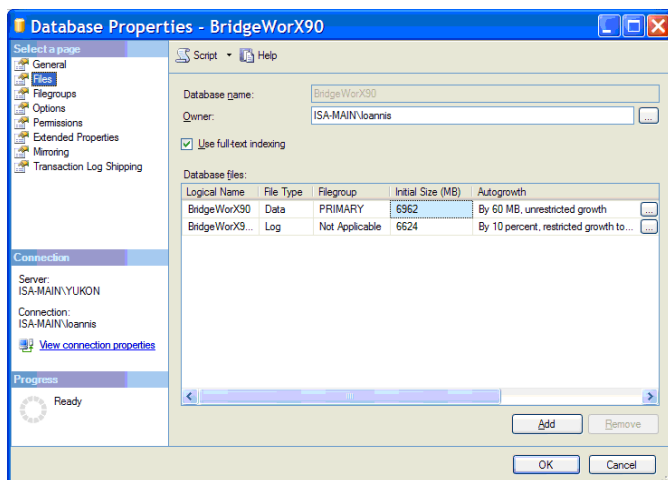


Figure 7 - Setting database size & growth options

Configure archiving of database information

Both BridgeWorX and ReportWorX have a built-in archiving mechanism that is processed transparently during runtime operations as a way of controlling the overall size of the BridgeWorX and ReportWorX databases. The configuration dialog can be accessed within the BridgeWorX or ReportWorX Configurators under the Tools > Options menu:

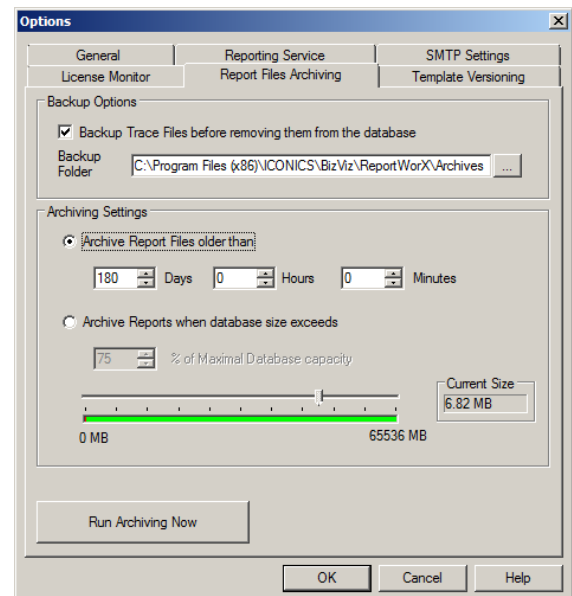


Figure 8 - Using built-in archiving support

The user can perform manual archiving tasks or set the automatic archiving settings. Once the user-configured criteria are met, the BridgeWorX or ReportWorX runtime services will periodically archive executed reports or transaction trace logs to the specified user folder and delete their information from the BridgeWorX or ReportWorX runtime databases. This built-in archiving scheme is a great tool for controlling BridgeWorX and ReportWorX database sizes, for small to medium applications.

In the case of BridgeWorX, if the end user application has a significant load of more than 1,500 transactions per day, then it is recommended that a native SQL Server maintenance job be created to control the size of the BridgeWorX database instead of using the built-in transaction archiving support. This job should be scheduled to run during downtime or low transaction activity intervals and it should be configured to maintain only the most recent transactions listed in the BWX_TRANS_INFO table of the BridgeWorX database, based on the Trans_Execution_Date field in that table. A query like the one below can effectively be used:

```
delete from BWX_TRANS_INFO where
Trans_Execution_Date <= DATEADD(day, -
7, GETUTCDATE ( ) )
```

This query would maintain only the transactions within the last 7 days. A similar query can be used for the ReportWorX configuration database, by substituting the corresponding RPTWX_REPORT_INFO table.

Miscellaneous Optimization Tips

Finally, here are some miscellaneous optimization tips that can have a significant impact on BridgeWorX or ReportWorX performance:

System paging file size (virtual memory settings)

It is recommended that the user sets the virtual memory settings to be 2 to 4 times the size of the physical RAM installed on the BridgeWorX or ReportWorX node. This will improve runtime performance as well as SQL Server memory utilization.

To get to this dialog:

- 1) Right-click on My Computer and select Properties.
- 2) Click on the Advanced tab.
- 3) Click the Settings button under the Performance category.
- 4) Click the Advanced tab on the subsequent dialog.

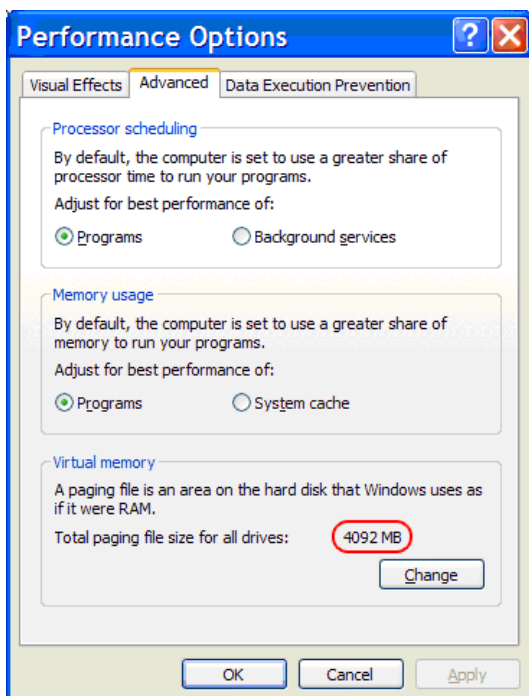


Figure 9 - Setting system virtual memory settings

SQL Server memory settings

If SQL Server is installed and running locally on the same node as BridgeWorX or ReportWorX it is suggested that the user configures the server not to use more than 50-75% of the total available system RAM. This can be set within the SQL Server Enterprise Manager.

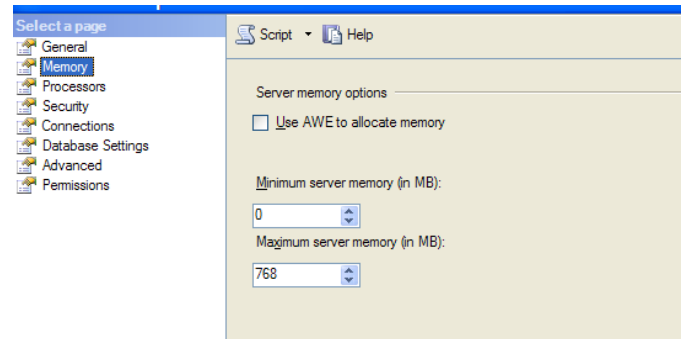


Figure 10 - Setting SQL Server memory settings

In general, ICONICS recommends monitoring your BizViz applications during initial application setup to detect any issues relating to system load or runtime performance. The Unified Web Interface displays report and transaction statistics, which can be used to monitor overall system runtime behavior. In addition, the Windows Performance Log can be used to monitor overall system resource utilization, and TraceWorX can be used to debug any communication issues.