

Optimizing iSeries SQL Performance

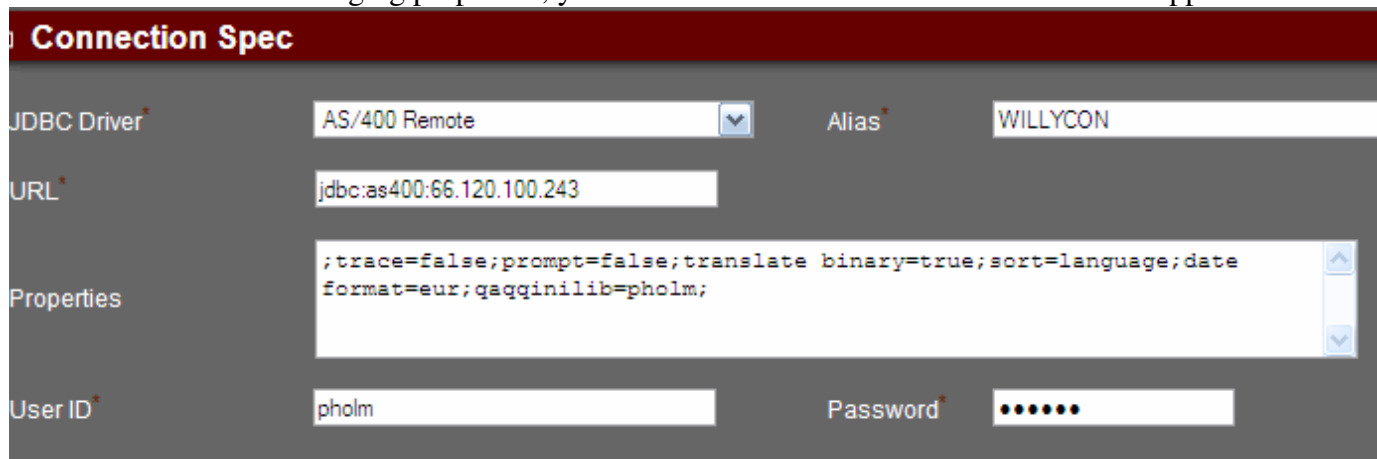
Introduction

This document describes some of the optimization tools and techniques related to WOW development on the IBM System I. WOW uses standard IBM interfaces and APIs. These interfaces include IBM's ODBC/JDBC interface which allows remote SQL to be performed. Since these interfaces are IBM supplied, all traditional work management and SQL optimization techniques are applicable and outside the scope of WOW's direct control.

The following items are supplied as a convenience for IBM System I customers.

JDBC connection properties:

WOW uses IBM's JDBC driver for SQL access. Many properties can be set the WOW connection as shown below. When changing properties, you must restart the connection for them to be applied.



The screenshot shows a 'Connection Spec' dialog box with the following fields:

- JDBC Driver:** AS/400 Remote
- Alias:** WILLYCON
- URL:** jdbc:as400:66.120.100.243
- Properties:** ;trace=false;prompt=false;translate binary=true;sort=language;date format=eur;qaqqinilib=pholm;
- User ID:** pholm
- Password:** (masked with dots)

- List of properties:
<http://publib.boulder.ibm.com/infocenter/systems/index.jsp?topic=/rzahh/jdbcproperties.htm>
- All properties are in the form of key=value and separated by a “;” (semi colon).
- Properties include date/time formats, performance, logging, etc

Controlling SQL Using the QAQQINI file:

A WOW SQL operation when executed is simply sent to IBM's SQL processing APIs. When execution starts the processing occurs totally within OS/400. Some SQL queries may result in long execution times that affect other users. IBM has supplied a database file to allow users to “control” different aspects of the SQL execution such as query timeout. Follow these instructions to setup your own query control file. IBM's supplied file is called QAQQINI.

1. Create a duplicate object using this CL command:

```
CRTDUPOBJ OBJ(QAQQINI) FROMLIB(QSYS) OBJTYPE(*FILE) TOLIB(QGPL) DATA(*YES)
```

2. Add a connection property to specify which library to locate the ini file to your WOW connection. Refer to the JDBC Properties section for an example. To use QAQQINI file in library QGPL, you would specify “ ;qaqqinilib=QGPL; “

3. To control SQL execution, modify the records in the file QGPL/QAQQINI. You can easily do this by created a WOW admin SQL operation such as: SELECT * FROM QGPL/QAQQINI . The field with the attribute setting is QQVAL. Each row contains a description of the attribute. In the following example, we have set the maximum query execution time to be 14 seconds for the attribute QUERY_TIME_LIMIT.

Details of each controllable setting can be found here:

<http://publib.boulder.ibm.com/infocenter/series/v5r4/index.jsp?topic=/rzajq/qryoptf.htm>

4. Restart the WOW connection and test.

WOW Web Object Wizard | iSeries QAQQINI Settings

QAQQINI Settings Zip Codes

Messages

1 Row(s) Updated.

	▲ Query option parameter ▼ ⚙	▲ Query option parameter value ▼ ⚙	▲ Query option text ▼ ⚙
	APPLY_REMOTE	*DEFAULT	Specifies for database queries involving distribut
	PARALLEL_DEGREE	*DEFAULT	Specifies the parallel processing option that can l
	ASYNC_JOB_USAGE	*DEFAULT	Specifies the circumstances in which asynchronou
	QUERY_TIME_LIMIT	14	Specifies a time limit for database queries allowe
	UDF_TIME_OUT	*DEFAULT	Specifies the amount of time, in seconds, that the
	MESSAGES_DEBUG	*YES	Specifies whether query optimizer debug messag
	PARAMETER_MARKER_CONVERSION	*DEFAULT	For dynamic SQL queries, specifies whether or no
	OPEN_CURSOR_THRESHOLD	*DEFAULT	Specifies the threshold to start full close of pseu

Any SQL that may take more than 14 seconds to execute would receive an exception and not execute as shown below.

QAQQINI Settings Zip Codes

Messages

[SQL0666] Estimated query processing time 24 exceeds limit 14. [SQL0666] Estimated query processing time 24 exceeds limit 14.

▲ Query option parameter ▼	▲ Query option parameter value ▼	▲ Query option text ▼
APPLY_REMOTE	*DEFAULT	Specifies for database queries involving distributed files, whether or not the CHGQRYA que
PARALLEL_DEGREE	*DEFAULT	Specifies the parallel processing option that can be used when running database queries
ASYNC_JOB_USAGE	*DEFAULT	Specifies the circumstances in which asynchronous (temp writer) jobs can be used to help
QUERY_TIME_LIMIT	14	Specifies a time limit for database queries allowed to be started based on the estimated r

Changing job priorities for QZDASOINIT Jobs:

WOW uses IBM prestared jobs for SQL access. These jobs have the name: QZDASOINIT and can be found using the CL WRKACTJOB command.

By default these jobs run at priority 20 which is the same as interactive jobs. You can control the priority and resource usage of these jobs. In this example, the priority is changed to 21, which is slightly lower than its default value of 20. This will reduce the impact to interactive users. Note that most interactive jobs run at priority 20.

1. DSPSBSD QSERVER

2. Verify that the status is Active. If not, use WRKSBSD *ALL/ QSERVER to find the active subsystem. Note the library (normally it's QSYS).

3. Substituting the library from step 2 in the FROMLIB parameter:

```
CRTDUPOBJ OBJ(QPWFSEVER) FROMLIB(QSYS) OBJTYPE(*CLS) TOLIB(QSYS)
NEWOBJ(QPWFJDBC)
```

4. Change the run priority on the new class: CHGCLS CLS(QSYS/QPWFJDBC) RUNPTY(21)

NOTE: Evaluate the other parameters on the CHGCLS command and set as appropriate for your organizations environment.

- 5. CHGPJE SBSDB(QSYS/QSERVER) PGM(QSYS/QZDAINIT) CLS(QSYS/QPWFJDBC)
- 6. CHGPJE SBSDB(QSYS/QUSRWRK) PGM(QSYS/QZDASOINIT) CLS(QSYS/QPWFJDBC)
 - 7. Depending on your system configuration, the subsystem may need to be restarted.
 - a. *ENDSBS SBS(QUSRWRK) OPTION(*IMMED)*
 - b. *STRSBS QUSRWRK*

Once changes are in place, any new QZDASOINIT jobs will run at the new priority (21). The following screen shot shows JDBC jobs lowered to priority 23.

```
Display Job Run Attributes
Job:  QZDASOINIT      User:  QUSER      Number:  949831
Run priority . . . . . : 23
Time slice in milliseconds . . . . . : 3000
Eligible for purge . . . . . : *NO
Default wait time in seconds . . . . . : 30
Maximum CPU time in milliseconds . . . . . : *NOMAX
CPU time used . . . . . : 610
Maximum temporary storage in megabytes . . . . . : *NOMAX
Temporary storage used . . . . . : 5
Maximum threads . . . . . : *NOMAX
Threads . . . . . : 1
```

Improving Query Performance:

There are a variety of methods for analyzing a query for performance. The below suggestions are for creating indexes that the queries would use:

- 1. Run STRDBG, run the query via STRQSL, then check the job log for suggestions.
- 2. For V5R4 and later, the iSeries Navigator has a Database option called the Index Advisor

References:

<http://www.redbooks.ibm.com/redbooks/SG246654/wwhelp/wwhimpl/js/html/wwhelp.htm>

Monitoring SQL Performance Using IBM Index Monitor

<http://www.redbooks.ibm.com/redbooks/SG247326/wwhelp/wwhimpl/common/html/wwhelp.htm?context=SG247326&file=3-6.htm>