

Out in Left Field

Volume IV, Issue V

IBM i for Business Database Performance

CONNECTING DB2 FOR i AND .NET

I find that a lot of work IT staff performs on a daily basis involves integrating disparate technologies. With vast data stores residing on the System i machines and the pervasiveness of Microsoft (MS) technologies in our offices, it is inevitable that at some point in time you'll need to have MS talk to i.

In my book, that translates to leveraging MS visual development tooling (Visual Studio 2008 or **VS2008**) for the GUI client development and **DB2 for i** for data retrieval and persistence.

Prerequisites

- (1) V5R3-V6R1 versions of the operating system.
- (2) First, install Microsoft's development, Visual Studio 2008 (VS2008).

Microsoft offers a free, Express version of Visual Studio. You can download it at this link:
<http://www.microsoft.com/express/download/default.aspx>

Depending on what you intend to develop with, you have choices of four programming languages, all with a Visual prefix: Basic, ASP.NET, C# and C++. Select all of them or just the ones that you intend to use (i.e. for Web development, only download ASP.NET).

Inside this issue:

Connecting DB2 for i and .NET	1
Sequential Thinking about Random I/O	2
CTO Memo	4
Budget Efficiency through Database Efficiency	4
Optimistic Locking and DB2 for i 6.1	6

I also recommend downloading SQL Server 2008 as well. This allows you to use the sample programs against Microsoft's Northwind sample database. By simply changing the connection string to a DB2 server, you can show that the same code works against DB2 data as well.

Of course, if you can afford it, install a full version of VS2008 instead of the Express edition.

- (3) Install the Data Access Components (DAC) for 6.1 version of IBM Client Access.

The easiest way to install DAC is to access the client access install image located on your 6.1 System i. In my case, the install image is located in the directory:

```
\\<your system  
name>\Q\IBM\ProdData\Access\Windows\Image32  
Setup.exe
```

Note that there are three versions of Windows install image: Image32, Image64a and Image64i. Since I wanted a 32 bit version install, I defaulted to the Image32 folder. Another method is to use the installation CDs shipped by IBM.

- (4) Load the latest service pack for your version of the Client Access.

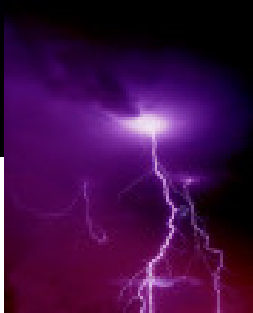
At the time of this writing, the latest 6.1 service pack is SI31390 and its details are included at this link:
http://www-912.ibm.com/s_dir/slkbase.nsf/ibmscdirect/404D5CF497B3C3B986257432001FBE5D
For V5R4, the latest is SI31388 and for V5R3 it is a SI30707 service pack.

- (5) Make sure "IBM DB2 for i5/OS .NET Provider" (I'll call it iDB2) shows up in VS2008 Server Explorer

Much too often this simple restriction trips up a novice .NET developer as well as an MS developer not experienced with the System i. Many of the samples use data access design time wizards (i.e. Table Adapter, Grid Control) which demonstrate how easy it is to create them using Server Explorer support. If the iDB2 data provider doesn't show up as a design time option, this becomes a (temporary) showstopper. The reason why it doesn't show up in VS2008 is that the DAC hasn't been updated yet for that version, but rather for VS2005. A slight development lag is understandable given the fact MS and IBM don't have joint development schedules (not possible even with the imagination of the most creative SciFi minds).

To ensure it does show up you'll need to edit the windows registry. Fear not, it's much easier than it sounds:

(Continued on page 8)

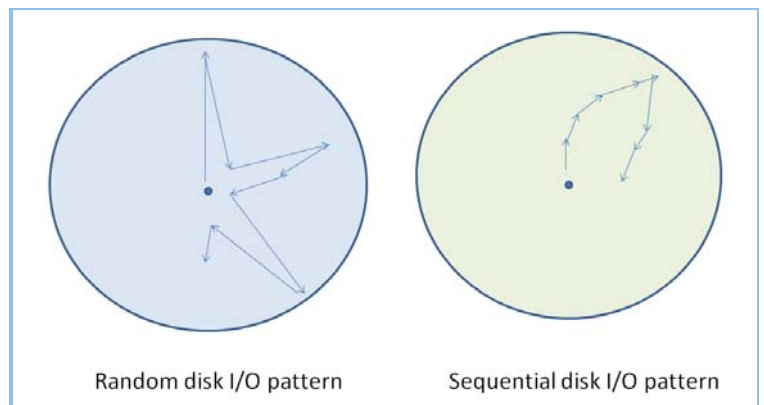


Sequential Thinking ABOUT RANDOM I/O

In our last newsletter, the topic of clustered data was explored. The objective of clustering is to reduce random disk I/O by putting rows of data in a specific key order. This presorted order allows the query optimizer (or your RPG application) to run faster because the time spent waiting to bring data into memory is reduced. If the query or the program retrieves a lot of data using the index used to sort the rows, the elapsed time of the query or batch job can be significantly shorter. Furthermore, other jobs on the system also benefit because the disk drives are less busy.

The diagram on the right illustrates what happens at the disk level when random I/O occurs. As can be seen on the left, the disk arm will move a much longer distance (on average) than it would if the next piece of data were known to be close. As can be seen on the right, because the row order is sorted the disk arm can sweep across the disk and get to the next element of data much more quickly because it has less distance to travel.

The downsides: At the risk of pointing out the obvious, data in a file or SQL table can only be in one order (i.e. sorted one way). This means that while you may use the reorganize



physical file member command to put the rows in a sorted order, queries or RPG programs that process the data in another order will see no benefit. Furthermore, it does take system resources to reorganize the data and if the files can't be reorganized "online", the associated application must be down for potentially a long period of time.

(Continued on page 11)



To begin with, the team here at Centerfield would like to thank Mike Cain and Tom McKinley for their excellent webinars on autonomic index advice and MQTs. As usual, Mike and Tom provided a very informative and lucid explanation of complex topics. And last month we hosted a webinar with IBMer Kent Milligan on the latest text search technology for i. Information can be found on our events page: www.centerfieldtechnology.com/events.asp While you're there, check out the recordings to previous events.

As both the weather and economy cool off, everyone is looking for a way to "keep warm." Hopefully we've provided some tips on this newsletter that will help you and your team accomplish your goals in this chilly environment.

If you are hunkering down with a tight budget, read the article on extending the life of your current system through greater database efficiency. If you have a powerful system, but are not yet happy with slow queries or how long batch jobs take, see the article on minimizing disk I/O with the

query optimizer. If part of your development strategy is to utilize Microsoft technologies and .NET, see Elvis' in-depth article on data providers and how to avoid some common technical issues.

As we all tighten our belts and wait for the temperature and the economy to warm up, let me remind you that Centerfield can help you get through this tough period with both our products and services. Please let us know how we can help!

On behalf of the team here in Rochester, we welcome all of our new customers this year and wish all of you – a safe and very happy New Year.

Best Regards,

Mark L Holm
Chief Technology Officer

Budget Efficiency through Database Efficiency

Two recent Centerfield consulting engagements were directly related to inefficient database access.

In one case, the customer had a fairly complex batch job that only used 1.5 to 3% of their system's CPU, but was not finishing before end-users wanted to start using the information. With some analysis of the call stack as the programs ran, it was apparent that very little time was being spent in the application code. In fact, over 95% of the elapsed time was spent in SQL and DB2 runtime routines.

Another customer, running a large ERP application, had allocated memory in a way that hurt database-intensive work and the setup of their data replication tool was overkill for their real business requirements. The net effect was that their system was running well below full CPU capacity, yet wasn't meeting their performance objectives. In both situations, these customers had very powerful machines but database work had become a bottleneck for their end-users, partners, and business processes.

The bottom line is that there are almost always good opportunities to better utilize hardware as well as extend the life of a system with either tools or a small amount of expert analysis. In both cases, the IT budget can be spared in these tough economic times.

Application Tuning for DB2

As systems have become incredibly powerful, applications are spending more and more time processing database requests and less time in application logic. As a result, a fundamental shift is necessary in how systems are tuned to efficiently run applications. Instead of worrying about elaborate subsystem configuration and esoteric work management settings, it makes much more sense to make database access more efficient. This is especially crucial with the de facto use of SQL for new applications and reporting tools. Furthermore, CPU is not generally the focus of performance tuning. Today, the critical metric is clock time.

So what options or tools are available to performance tune the System i to be the best application database or database server it can be? In the grid on the following page, we've provided a partial list of the critical factors necessary to achieve good database performance. We've also listed which of these factors are examined during a Centerfield consulting engagement as well as the capabilities our HomeRun suite provides.

(Continued on page 4)



Critical factor	Technical description	Centerfield's role
Memory utilization	Examine pool configuration including sizes, distribution and use of expert cache.	Consulting
Pre-started job configuration	Look at prestarted job configuration for QZDASOINIT and QSQRVVR jobs to reduce startup overhead.	Consulting
SQL statements	Identify the worst performing statements and why they are inefficient. Involves capturing statements as they run and analyzing their performance to find the key issues and correct them.	HomeRun & Consulting
Index recommendations	Examine the database and SQL to determine if additional indexes would improve the performance of SQL statements.	HomeRun & consulting
Compilation options	Determine if compiler options are resulting in subpar performance of queries	Consulting
Unused indexes	Identify truly unused indexes (not those “secretly” used by the query optimizer) and recommend a safe approach to reducing their overhead for highly changed files and tables	HomeRun
Sub-optimal indexes	Find SQL indexes and Logical files that could be improved in some way to boost the performance of applications and queries which use them	HomeRun
Redundant indexes	Identify duplicate indexes that are likely not necessary	HomeRun
Use of the old and new query optimizer	Diagnose why the old query optimizer is being used and identify ways to ensure the new (faster) query engine is used instead	Consulting
Database-related System values	Look at system values that influence query optimization and recommend changes to fit the application and system environment	Consulting
Journaling analysis	Examine the journaling configuration and identify potential bottlenecks related to disk I/O or CPU use and recommend changes to streamline it	Consulting
Use of the QAAQINI file	See if the query options file is being used and why. Recommend changes to fit application and database performance objectives.	Consulting
System managed access path protection	Identify opportunities to optimize the use of SMAPP especially in high availability environments	Consulting
Data type analysis	Look at the use of certain data types to ensure they are defined to minimize I/O while minimizing disk space	Consulting
ODBC/JDBC data source configuration	Examine the options used by these functions and determine if the optimal settings are being used.	Consulting
Joblog messages	Look at jobs running SQL to ensure they are not getting messages that would indicate either performance or functional issues	Consulting

As you can see even from this partial list, there are quite a number of factors that influence the efficiency of your application's use of DB2. This is especially true if the application is SQL-based, large, or contains many hardware or software layers. As you consider your financial plan in the next year, keep in mind the possible options and remember Centerfield can help you can achieve budget efficiency through database efficiency.

What would a 10 or 20% performance improvement mean to your bottom line?



Optimistic locking and DB2 for i 6.1

There are two general strategies for locking database rows.

A pessimistic locking strategy assumes that the probability is high another user will try to modify the same row in a table that you are changing. In RPG and other HLLs on the System i, a developer may implement a pessimistic method by locking the record to ensure no one can access it until an update is performed and the record released. In SQL, to avoid conflicting concurrent updates, the time tested solution is to lock the row using commitment control with an isolation level of repeatable read or even higher (i.e. table lock). Commitment control requires journaling be started for the table.

The advantage of pessimistic locking is that it guarantees changes are made consistently and safely. The disadvantage is that locks may be held for a long time and affect the throughput of the application with multiple users. Yet another disadvantage is that journaling is required, which incurs additional performance overhead on the application. Finally, and most serious disadvantage, is that a stateful connection is required. This last requirement is a showstopper for many web applications.

An alternative solution addressing concurrent updates of the same data is the optimistic locking strategy. Optimistic locking assumes that it is highly unlikely that concurrent updates will occur, and if they do, there is a mechanism in place to retry the transaction by redisplaying the newly updated data to the end user and giving them a chance to restart the transaction. The advantage of this strategy is that it is more scalable, does not require journaling, performs well and is relatively easy to understand and implement. The disadvantage is the additional retry logic needs to be instrumented in the application to handle the rare case where concurrent updates occur.

As you've probably guessed, optimistic locking is a preferred solution for any stateless web applications. Since it fits the stateless application model so well, optimistic locking has become very popular within the web development culture.

Today, I will focus on the implementation of the optimistic locking strategy for an application using SQL to access the System i data (as is the case with most Web applications). In addition, since the optimistic locking concept is not new on the System i, I will also mix up some of the cool DB2 for i 6.1 enhancements that standardize the way optimistic locking can be implemented.

Optimistic locking implementation

Quite simply, one implements the optimistic locking concept by having a smarter UPDATE and DELETE logic. Instead of doing something like:

```
UPDATE myTable SET (f1,f2,f3) = (newValue1,newValue2,newValue3)
WHERE f1 = oldValue1 AND f2 = oldValue2 AND f3 = oldValue3
```

the application would do something like:

```
UPDATE myTable SET (f1,f2,f3) = (newValue1,newValue2,newValue3)
WHERE f1 = oldValue1 AND f2 = oldValue2 AND f3 = oldValue3
AND updateTimestamp = oldTimestampValue
```

Notice the addition of the timestamp selection criteria to the WHERE clause of the update statement. If another job had updated the row since it was read and displayed to the end user, this UPDATE would fail. With this implementation, it is crucial that the application checks the SQLCODE or SQLSTATE and if not successful, perform a retry with the new values displayed or aborts the entire transaction.

How does 6.1 help

So far, I haven't really shown you anything new, so let me get to the 6.1 enhancements that make implementation of optimistic locking a breeze. As you know, blocked inserts and speed of today's boxes make it possible for a timestamp to not be unique. This could play havoc on the optimistic locking design. Developers solved it by adding yet another criterion to the UPDATE, a unique identifier. This solution is satisfactory, but complicates optimistic locking implementation a bit more.

With DB2 for i 6.1, we have two new items that can make a developer's life easier:

- implicitly hidden columns
- auto-generated timestamp

(Continued on page 7)

(Optimistic locking and DB2 for i 6.1—Continued from page 6)

Hidden columns

Let's say I create a table:

```
CREATE TABLE myTable (f1 INT, f2 CHAR(10), f3 TIMESTAMP IMPLICITLY HIDDEN);
```

If I then tried to INSERT a row into a table using the traditional and common syntax:

```
INSERT INTO myTable VALUES(1,'Row1',CURRENT_TIMESTAMP);
```

I would get an SQL0117 exception message with text "Statement contains wrong number of values". This means that I'm trying to insert into first two columns, and since the third has no default value defined, I also have to include it in the insert. The implicit "*" or ALL identifier's meaning has changed to only include the first two columns, due to my use of IMPLICITLY HIDDEN keywords in the column definition.

To correct my INSERT, I will explicitly spell out the columns that I intend to insert into:

```
INSERT INTO myTable (f1,f2,f3) VALUES(1,'Row1',CURRENT_TIMESTAMP);
```

When I then run a:

```
SELECT * FROM myTable
```

I only get the first two columns back, again illustrating the point that the IMPLICITLY HIDDEN column is not included in the column list by default.

Since I really need to know the value of this column for the optimistic locking implementation to work, I will alter my SELECT statement slightly and include the hidden column in the result set:

```
SELECT f1,f2,f3 FROM myTable
```

Hidden columns by themselves are interesting, but don't really bring us home in regards to the optimistic locking implementation. So let's move on to another 6.1 enhancement that completes the requirement for optimistic locking.

Auto-generated timestamp

Let's alter my table creation script slightly:

```
CREATE TABLE myTable (f1 INT, f2 CHAR(10),
f3 TIMESTAMP NOT NULL IMPLICITLY HIDDEN
FOR EACH ROW ON UPDATE AS ROW CHANGE TIMESTAMP);
```

Now, let's try my simple INSERT that failed in the original example:

```
INSERT INTO myTable VALUES(1,'Row1')
```

It worked! The reason it succeeded this time is that we told the database manager to auto-generate it for us.

The SELECT statement from the previous example (SELECT f1,f2,f3 FROM...) still behaves the same and returns the following output:

F1	F2	F3
1	Row1	2008-11-11-16.58.59.706663

Notice what happens when I run the UPDATE statement though:

```
UPDATE myTable SET (f1,f2) = (2,'Row2') WHERE f3 = '2008-11-11-16.58.59.706663'
```

If we display the row again, we'll notice that the timestamp has changed:

F1	F2	F3
2	Row2	2008-11-11-17.00.21.871014

The reason it changed is that DB2 has automatically updated it due to our column definition keywords at CREATE TABLE time. The nice thing about this type of timestamp column is that it is **guaranteed** to have a unique value, even in blocked inserts.

Bringing it all together

We now have all the pieces to enable a Web developer developing stateless web applications to leverage optimistic locking implementation when altering our database.

A Database architect can create a new table or alter an existing one to include an auto-generated, implicitly hidden timestamp column. The Web developer's SELECT statement needs to include all columns of interest, including the implicitly hidden timestamp column. When the end user is ready to complete his web transaction, the UPDATE statement needs simply to include the following clause:

```
... WHERE autoGeneratedTimestamp = oldTimestampValue
```

Further, application needs to verify that the UPDATE succeeded. If not, redisplay the page with new values and ask the user to restart the transaction. As previously mentioned, the retry scenario should occur very rarely in the optimist locking scenario.

Conclusion

Optimistic locking is a very popular concept in the world of stateless web development. Many System i web developers are already implementing it one way or another. The new DB2 for i 6.1 enhancements of implicitly hidden columns and auto-generated timestamps make the optimistic locking implementation a breeze.

(Connecting DB2 for i and .NET— Continued from page 1)

- a) Export the key for "IBM DB2 for i5/OS .NET Provider":
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\VisualStudio\8.0\DataProviders\{409DB050-2D37-4165-BEDF-26791FFDF728}]
- b) Right-click on the newly exported *.reg file and Edit it
- c) Do a mass Search/Replace all of the "\8.0\" instances with "\9.0\" instances
- d) Save your changes and Exit
- e) Double-click the edited *.reg file and import it back in the registry (it'll automatically go into the correct Data Provider registry key)
- f) Export the key for "IBM DB2 for i5/OS" data source:
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\VisualStudio\8.0\DataSources\{61DEDEB0-AC66-49e7-B2DD-605D3B148B6E}]
- g) Right-click on the newly exported *.reg file and Edit it
- h) Do a mass Search/Replace all of the "\8.0\" instances with "\9.0\" instances
- i) Save your changes and Exit
- j) Double-click the edited *.reg file and import it back into registry (it'll automatically go into the correct Data Source registry key)

Trust me, this simple step will make your life as a novice Microsoft developer **much** easier.

Lest I forget, thanks to DaytonaDave for making me aware of this workaround.

Using Server Explorer

Server Explorer is an inherent part of VS2008. Since any application touching the System i has a 99.99% probability of accessing database data, knowing which data access provider to use is critical. The good news is that IBM offers a lot of options.

The first option and one recommended by IBM, is to use the iDB2 provider (its .NET namespace is *IBM.Data.DB2.iSeries*). It has been developed with the System i and .NET in mind specifically and ought to perform the best out of all available options.

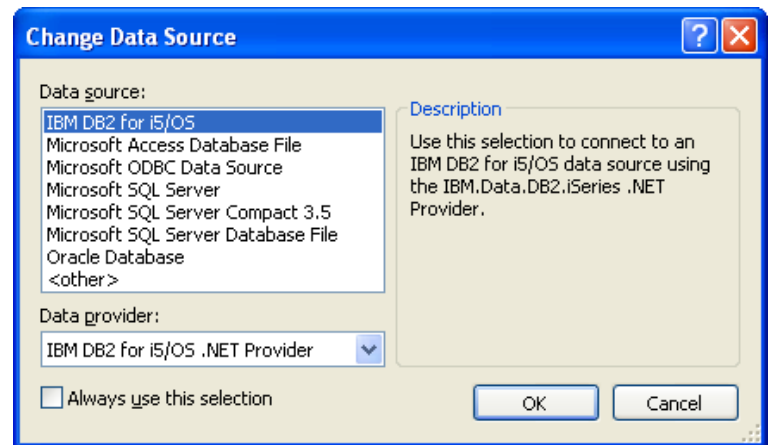
Whether the performance claim is true or not will depend on the source and your runtime environment. I do like the fact that the claim comes from the Rochester Lab as well as the fact that iDB2 is continuously supported and improved by the System i developers. Accordingly, I too recommend using the iDB2 data provider whenever possible and I am going to lead with it as the number one option for data providers.

As an exercise examining different providers, let's create a number of different connection strings, which you can then use in any number of Microsoft and 3rd party data access design time wizards.

- (1) Start a new project in VS2008

- (2) Click on **View → Server Explorer** to ensure the Server Explorer toolbox is easily available to you as a docked panel
- (3) In Server Explorer, right-click on the **Data Connections → Add Connection → Change**

You will see the list of available data sources:



Notice that due to our handy registry import, the iDB2 provider now shows up as a default option!

- (4) Click **OK**
- (5) Change the Data Source option to your database (server) name and the Default Collection to the production library (libraries) you would like to access by default. There are a number of other options you can change as well (naming convention, library list etc.).
- (6) Click **Test Connection** to verify options you've typed in are valid
- (7) Click **OK** to confirm that Test Connection worked
- (8) Click **OK** to finish creating the Data Connection string. You now see a new connection string show up in the Data Explorer that matches up your database (server) name.

Now that we've created an iDB2 data source, let's see how easy it is to create connections strings that leverage other IBM data providers.

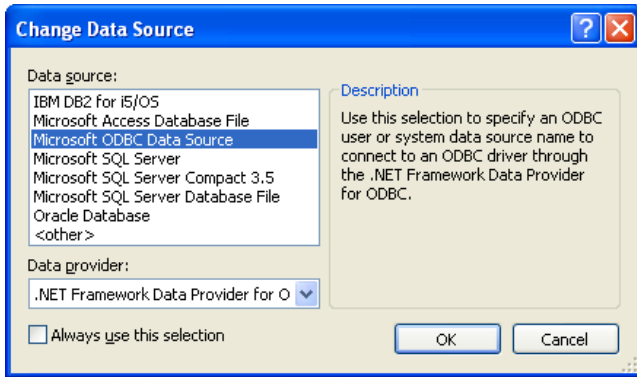
Let's continue our examination of the remaining data providers with a look at the ODBC data provider:

- (1) First, create an ODBC data source provider using the ODBC Administrator facility. You can start it through a windows Control Panel, or using this shortcut:
Click on **Start → Run → odbcad32 → OK**
- (2) Click on **Add → Client Access ODBC Driver (32 bit) → Finish**
- (3) Select the appropriate options for server, naming, library list etc. and name it something like *ODBCDataProvider*
- (4) Click **OK → OK**
- (5) Back in VS2008 Server Explorer, right-click on **Data Connections → Add Connection → Change**

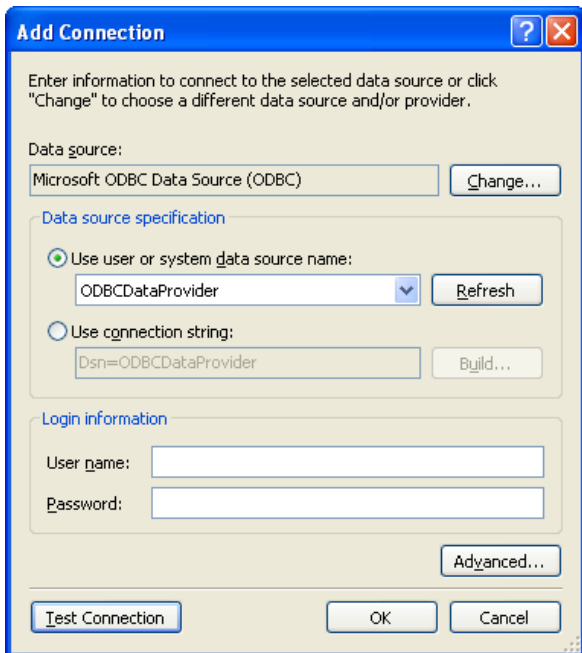
(Continued on page 9)

(Connecting DB2 for i and .NET— Continued from page 8)

(6) Select the Microsoft ODBC Data Source as such:



- (7) Click **OK**
- (8) Select ODBCDataProvider from the “Use user or system data source name:” dropdown



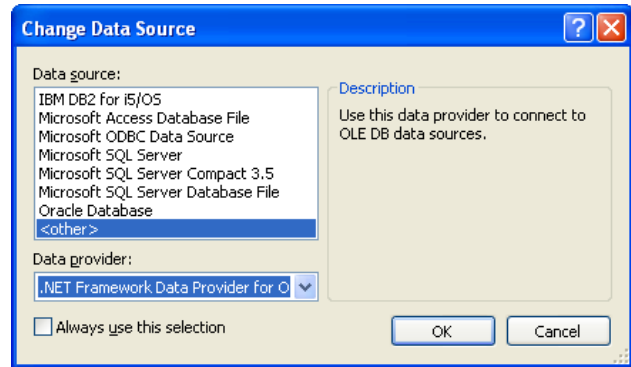
- (9) Test the connection again and click **OK** to finalize the process
- (10) You will see a new connection string in the Server Explorer that is by convention titled like: **ODBC.SERVER.DATABASE** name.

ODBC is a stable technology and even though it runs “unmanaged code”, it remains a valid choice for a data provider.

IBM also offers 3 different OLEDB data providers. Here are steps to creating connection strings for them:

- (1) As usual, begin with the right-click on **Data Connections** → **Add Connection** → **Change**

- (2) Select the **<other>** ‘Data source’ **AND** the “**.NET Framework Data Provider for OLE DB**” ‘Data provider’, as such:



- (3) Click **OK**
- (4) Invoke a dropdown selection on “OLE DB Provider:” dropdown box and select one of 3 available OLE DB provider: **IBMDASQL, IBMDARLA, IBMDA400**
- (5) Type in Server name
- (6) Type in or select the database name you want to use in the “Initial catalog:” dropdown
 NOTE: Most of the time the database name on System i matches the server name (default configuration). A system administrator may opt to give database a different name though, so you’ll need to know what it is in that case.
 On top of that, each Independent Auxiliary Storage Pool (iASP) will have a unique database name as well.
 Choose accordingly.
- (7) Test the connection again and click **OK** to finalize the process
- (8) You will see a new connection string in the Server Explorer that is by convention titled like: **<OLE DB provider name>.SERVER.DATABASE** name.

You can go through the same steps for all 3 OLEDB providers and create 3 different OLEDB connection strings which you can subsequently leverage in your code or design time data access wizards.

Finally, let me also mention that there is yet another true DB2 .NET data provider you can use. It’s the one included with the DB2 LUW and DB2 Connect distribution. The steps of creating a connection string for it should match the steps outlined for iDB2 provider.

However, I’m not going to show you those steps as the only way for me to get my hands on the DB2 LUW provider that works with System i requires installation of **DB2 Connect** and in the interest of time, I elected not to take that step.

That brings our total number of IBM offered .NET data providers to 6. One, or some combination of them, ought to suit your needs.

(Continued on page 10)

(Connecting DB2 for i and .NET— Continued from page 9)

Data Provider choice

So, what provider should you use? I'm afraid the answer is easy to say, but difficult to work with: *IT DEPENDS*.

If performance is your number one priority, some **non-scientific** testing shows that iDB2 provider is the best choice, with OLEDB IBMDASQL choice being a very close second. The ODBC provider came in last with DB2 LUW provider preceding it. If performance is not the prevailing factor, your decision is a bit more difficult to make as each provider varies slightly in the function it offers.

For example, the DB2 LUW with DB2 Connect provider has the best VS2008 integration. You even get an additional VS2008 toolbar focused solely on IBM services offered by the DB2 Connect. DB2 Connect of course is meant to be a multiplatform interface that works with all flavors of DB2, System i and System z (mainframe) included. If you are a large shop with a variety of servers and DB2 databases, DB2 Connect is probably a perfect choice for you.

The ODBC provider may be attractive for clients that want to avoid use of managed (read .NET) code and prefer the time-tested ODBC unmanaged code (read Windows binary executable code). Use of Extended Dynamic support (SQL Packages) may also be a key factor in your choosing the ODBC data provider.

OLE DB providers have been very popular with the development community, for a variety of reasons. IBMDASQL SQL-only provider supports Commitment Control and MTS (Microsoft Transaction Services). IBM DARLA RLA-only (Record Level Access) supports true record blocking and forward-only cursors for record level access. IBM DAA400 in addition to SQL and RLA, also offers access to remote command and program calls as well as data queues.

The wonderful thing about data access in .NET environment is that you can easily shift between different data providers within your code and leverage the unique functions offered by each one of these providers. In other words, use whatever provider will help you get your work accomplished.

My take on the data provider selection is to go with iDB2 provider wherever possible and only revert to one of the other providers when you need to use a function not supported by the iDB2 provider.

Source example, I crave an example

I could give you an example, but I'm not going to. Instead, a multitude of examples can be found in the Microsoft documentation or with a Google-search. All you have to do is plug in an iDB2 connection string instead of the one showed in the Microsoft example.

If you're opposed to looking at Microsoft examples, consider downloading the "Integrating DB2 Universal Database for iSeries with Microsoft ADO .NET" Redbook available at this link: <http://www.redbooks.ibm.com/abstracts/sg246440.html>

I have used this Redbook for parts of this article and it has some nice examples integrating with the grid control (i.e. 3.3 and 4.6.1 sections).

Conclusion

Whether you're a seasoned System i developer venturing to the .NET realm or a seasoned .NET developer venturing into the System i realm for the first time, the goal of my article is to help you overcome the initial tripping points.

Happy coding.



(Sequential Thinking about Random I/O— Continued from page 2)

The alternative: The good news is that the query optimizer can reduce random disk I/O on its own with a technique called “skip sequential.” Before we discuss what skip sequential means, let me talk about the use of an index (or logical file) in the traditional way.

When a key value is passed to DB2 for a search (e.g. key equal) the database engine looks that key value up in the index. Once the key is found, the row associated with that value is located on disk, brought into memory, and then passed back to the application or query. If this process is done in a loop, it is likely that the job will do many random I/Os. This is because there isn't a direct relationship between a key value and the associated record's location on disk – thus the term “random I/O.”

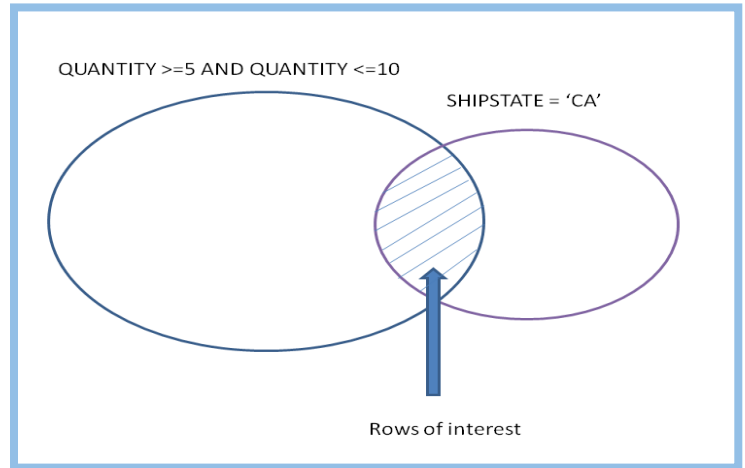
If an index is read sequentially the pages of the index will be brought into memory and referenced many times. While the index pages are used to find many different rows, the rows themselves may be spread out over a wide area. If the data has been sorted there is what is called “locality of reference.” In other words, the pages in the memory pool are used over and over again because when one page of data was brought into the pool it contained data that was going to be needed next.

Skip sequential: Ideally, data would always be read sequentially. On the other hand, it is important to use indexes to filter data so table scans are not necessary. This is where the concept of skip sequential helps the query optimizer. Let's say for example that a query is submitted to the system and an index can be used to identify the rows of interest. After looking at the index however, the optimizer guesses that approximately 35% of the rows will be selected. Normally, if the percentage is larger than 20% the optimizer will perform a table scan because it would be faster than using an index because of the random I/O.

In the case of a query that returns 35% of the data, the optimizer may decide that it is most efficient to use the skip sequential technique to find the rows. In that case, the index will be used to locate all of the keys and rows that pass the selection criteria. Once those keys have been gathered up, the list of relative record numbers (RRNs) can be sorted. Once the RRNs are in order, DB2 can then do a selective table scan and only request ranges of data be bought into memory that include the rows of interest. In doing that, the disk arms will sweep across the drives and avoid the back-and-forth movement that happens if the request was truly random.

Bit maps and skip sequential: What if one index isn't sufficient to determine the list of rows needed for a skip sequential? The answer is that the optimizer can also use a technique called index anding/oring to achieve the same goal. Let's say for example you have the following query:
`SELECT * FROM ORDERS WHERE QUANTITY >= 5 and QUANTITY <= 10 AND SHIPSTATE = 'CA'`

If the optimizer examines the index over QUANTITY, it may find that 60% of the orders have between 5 and 10 items. In turn, the optimizer may find that 30% of the orders come from the state of California by looking at the index over the SHIPSTATE column.



In this case, the index over QUANTITY and the index over SHIPSTATE can be combined. First the keys that meet the QUANTITY criteria are gathered. Then the keys that meet the SHIPSTATE criteria are gathered. Both lists of RRNs sorted. Once they are sorted the optimizer can easily determine which RRNs are in both lists. The list of common RRNs are then isolated and used with the skip sequential process.

As a practical matter, the optimizer may choose to convert the list of RRNs into a “bit map.” A bit map is simply a different method to hold the record numbers. Instead of taking 4 bytes or more to hold each record id, a single bit can be used. The location of the bit in the string represents its relative position. In other words the second bit if the bitmap represents relative record number 2 and so on. If the query is complex and there are many indexes involved in the AND or OR operations, using bitmaps will often make sense because they can be easily combined using built-in machine instructions that are very fast and efficient.

Summary: Clustering rows in key order is the optimal way to access data accessed sequentially with a given index. Given that there can only be one “clustered index” however, it makes sense that the query optimizer has other clever techniques to leverage indexes while avoiding random I/O.

As a final note, here are some guidelines that will help the query optimizer use skip sequential:

- Make sure the query optimizer knows your query is read only – use the FOR FETCH ONLY clause.
- With embedded SQL use the ALWCOPYDTA (*OPTIMIZE) setting.
- Build single-key indexes to start. These are more likely to be combined using the index ANDing/ORing method described above.
- Consider the use of Encoded Vector indexes. This type of index is naturally suited to index ANDing/ORing and thus the use of skip sequential access.

Hopefully this article has provided some insight into yet another way of reducing the disk I/O bottleneck on your system. Good luck!

High-end paper maker improves customer responsiveness with a Centerfield Technology tuning solution for the IBM DB2 database



The truck is leaving the warehouse in 20 minutes and you need an up-to-the-minute report from your transactional database to be sure of what's supposed to be on the truck and where it's supposed to be going.

Or you're about to hit the "start" button on a manufacturing run to produce 50 tons of paper for a company's annual report and the customer suddenly calls for a change in the width of the paper. Or the color. Or the grade. Or the quantity.

Accommodating such last-minute change requests and delivering such fast enterprise reporting is beyond the scope of many paper manufacturers. But not at Mohawk Fine Papers, Inc., which has carved a niche for itself by virtue of agility and flexibility in meeting customer requests.

While many of its competitors have been acquired or disappeared in recent years, Mohawk has about doubled its own size through acquisition.

Headquartered in Cohoes, New York, the company has five facilities in New York and Ohio that annually deliver more than \$300 million in high-end paper products to distributors globally.

Special tuning

The abilities of Mohawk were amplified in mid-2008 with a special tuning solution for the IBM DB2® for i software (the relational database management system formerly known as DB2 for i5/OS®) that basically runs Mohawk's enterprise resource planning system.

The solution was provided by Centerfield Technology, Inc., an IBM Business Partner founded in 1997.

That solution enabled Mohawk to reduce, by up to several hundred percent, the time it took to get reports on the status of various mill processes — down to two minutes from 15 minutes on some reports, and down to five seconds from 45 seconds on others, said Paul Stamas, vice president of information technology at Mohawk.

IBM Business Partner:
Centerfield Technology, Inc.
Centerfield Technology provides DB2 for i and SQL optimization tools that help companies more effectively utilize their applications running on the IBM AS/400®, iSeries, and System i5® (System i®) computing platforms. Centerfield Technology is headquartered in Rochester, Minnesota.



Why were such time differences important? "Because our ability to take orders today and ship orders today and respond to late-breaking special orders from customers is what enables us to succeed in the premium paper segment," Stamas explained.

"It was a game changer. We're leveraging the power and reliability of the DB2-on-iSeries environment better than we ever did before."

Paul Stamas,
vice president,
information technology,
Mohawk Fine Papers

“Ninety percent of our orders are received and shipped the same day,” he said. Most large, integrated paper mills (i.e., those that own their own forests and pulp factories) set up their production runs to go for days or even weeks at a time, “but we change out our paper machines seven or eight times a day,” Stamas explained.

Mohawk also has a small transportation window of only a couple hours each day, where it needs to ship most of the orders. “Real-time reporting is all about accurate, timely information exchange between the sales offices and the shop floors and the loading docks. For us that’s imperative,” Stamas said.

To maintain this competitive edge and grow, Mohawk executives realized a couple of years ago that their flexibility and agility would have to reach new heights. But getting to that point was something of an odyssey.

Doubts resolved

“As we were looking to get closer to real-time reporting on plant operations,” Stamas said, “we had doubts that DB2 could reach that goal while it was simultaneously running our basic transactional system. DB2 powers our enterprise. We use it for taking orders, generating manufacturing instructions, ordering materials, arranging shipping, invoicing — the whole transactional workflow. Its robustness and reliability had long ago made it our primary repository for our enterprise systems.

“That said,” Stamas continued, “we were thinking of trying to reach our real-time reporting goals by offloading the reporting function to a Windows/Intel processor-based infrastructure, even though we knew it would be expensive, hard to accomplish and hard to maintain.”

But just before attempting that route, Vince Brower, manager of the Mohawk

enterprise computing platform (an IBM iSeries Model 520), happened across a Web-based seminar on high-performance data access. Conducted in March 2008, the Webinar was one of a continuing series of tutorials for IT professionals conducted by Centerfield Technology. Brower soon started conversing with Centerfield, and by mid-July, Centerfield had installed its HomeRun suite of database tools at Mohawk.

“We saw dramatic improvements immediately,” said Stamas. “It was a game changer. We’re leveraging the power and reliability of the DB2-on-iSeries® environment better than we ever did before.”

Optimized Partner

Centerfield participates in IBM PartnerWorld® Industry Networks, which offers a rich set of benefits to all IBM PartnerWorld members who want to team with IBM to build their vertical market capabilities, expand their partner network and attract customers in the markets they serve. Centerfield is “optimized” in the manufacturing sector, which means it has developed further specialization by optimizing its applications with IBM technologies, achieving success with its own solutions and other criteria.

Jennifer Halverson, director of IBM i Client Partners at Centerfield, said being an IBM Business Partner is an important part of the Centerfield marketing strategy.

“Our products have been highlighted in the IBM Business Partner Application Showcase,” she said. “We’ve used the lead-generation service. We’ve done direct mail and telemarketing with IBM, and we’ve hosted some of our customer meetings at the IBM Innovation Center in Chicago. It’s all been very positive for our business.”

For more information

Please contact your IBM sales representative or IBM Business Partner. Or you can visit us at: **ibm.com**

For more information about ISV resources from IBM PartnerWorld, visit: **ibm.com/partnerworld/industrynetworks**

For more about Centerfield Technology, visit **centerfieldtechnology.com**

For more about Mohawk Fine Papers, visit **mohawkpaper.com**



International Business Machines 2008
IBM Corporation
Route 100
Somers, NY 10589
U.S.A.

Printed in the United States of America
11-08
All Rights Reserved

International Business Machines Corporation, the IBM logo, ibm.com, AS/400, DB2, i5/OS, iSeries, PartnerWorld, System i, and System i5 are trademarks or registered trademarks of IBM Corporation in the United States, other countries, or both.

Other company, product and service names may be trademarks or service marks of others.

IBM hardware products are manufactured from new parts or new and used parts. In some cases, the hardware product may not be new and may have been previously installed. Regardless, our warranty terms apply.

References in this publication to IBM products or services do not imply that IBM intends to make them available in all countries in which IBM operates.

The information in this document concerning non-IBM products was obtained from the suppliers of those products or from their published announcements. IBM has not tested those products and cannot confirm the accuracy of the performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

IBM provides information technology products and services. In no event shall IBM's provision of products or services constitute legal advice and IBM does not represent or warrant that its services or products will guarantee or ensure compliance with any law. IBM's customer is solely responsible for identifying, interpreting and ensuring its compliance with applicable federal, state and local laws, and regulations and rules relevant to its business and should seek competent legal advice as necessary. Printed in the United States on recycled paper containing 10% recovered post-consumer fiber.