



97

DB2

November 2000

In this issue

- 3 Using Java with DB2
 - 11 All those joins
 - 18 Making sure that Business Intelligence aids intelligent business with DB2 Query Patroller – part 2
 - 26 DB2 Versions 6.2 and 7
 - 35 DRDA connectivity for DB2 UDB for OS/390
 - 48 DB2 news
-

© Xephon plc 2000

update

DB2 Update

Published by

Xephon
27-35 London Road
Newbury
Berkshire RG14 1JL
England
Telephone: 01635 38342
From USA: 01144 1635 38342
E-mail: trevore@xephon.com

North American office

Xephon
PO Box 350100
Westminster, CO 80035-0100
USA
Telephone: 303 410 9344

Subscriptions and back-issues

A year's subscription to *DB2 Update*, comprising twelve monthly issues, costs £255.00 in the UK; \$380.00 in the USA and Canada; £261.00 in Europe; £267.00 in Australasia and Japan; and £265.50 elsewhere. In all cases the price includes postage. Individual issues, starting with the January 1997 issue, are available separately to subscribers for £22.50 (\$33.50) each including postage.

DB2 Update on-line

Code from *DB2 Update* can be downloaded from our Web site at <http://www.xephon.com/db2update.html>; you will need the user-id shown on your address label.

Editor

Trevor Eddolls

Disclaimer

Readers are cautioned that, although the information in this journal is presented in good faith, neither Xephon nor the organizations or individuals that supplied information in this journal give any warranty or make any representations as to the accuracy of the material it contains. Neither Xephon nor the contributing organizations or individuals accept any liability of any kind howsoever arising out of the use of such material. Readers should satisfy themselves as to the correctness and relevance to their circumstances of all advice, information, code, JCL, and other contents of this journal before making any use of it.

Contributions

Articles published in *DB2 Update* are paid for at the rate of £170 (\$260) per 1000 words and £100 (\$160) per 100 lines of code for the first 200 lines of original material. The remaining code is paid for at the rate of £50 (\$80) per 100 lines. In addition, there is a flat fee of £30 (\$50) per article. To find out more about contributing an article, without any obligation, please contact us at any of the addresses above and we will send you a copy of our *Notes for Contributors*, or you can download a copy from www.xephon.com/contnote.html.

© Xephon plc 2000. All rights reserved. None of the text in this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, without the prior permission of the copyright owner. Subscribers are free to copy any code reproduced in this publication for use in their own installations, but may not sell such code or incorporate it in any commercial product. No part of this publication may be used for any form of advertising, sales promotion, or publicity without the written permission of the publisher. Copying permits are available from Xephon in the form of pressure-sensitive labels, for application to individual copies. A pack of 240 labels costs \$36 (£24), giving a cost per copy of 15 cents (10 pence). To order, contact Xephon at any of the addresses above.

Printed in England.

Using Java with DB2

Just about everyone has at least heard of Java and how it is going to transform the world of IT. But just because you've heard about Java doesn't mean you fully understand it. And even if you know a bit about Java, there is always more to discover. Let's face it, there's a lot of hype out there regarding anything that concerns the Internet. And now that DB2 for OS/390 fully supports Java, that hype will surely increase for those of us using the mainframe. Is it all hype or is there some hope for a brighter multi-platform world? Let's examine what Java means to the world of DB2.

WHAT IS JAVA?

First and foremost, Java is an object-oriented programming language. Developed by Sun Microsystems in 1991, Java was modelled after, and most closely resembles, C++. But Java requires a smaller footprint and eliminates some of the more complex and error-prone features of C and C++ (such as pointer management and the goto construct). Additionally, many tasks have been moved from the programming language to the JVM (Java Virtual Machine).

Java enables interaction with the World Wide Web. Although Web interaction is one of its most touted features, Java is a fully-functional programming language that can be used for developing general-purpose programs independently from the Web. But let's look at the Web aspect of Java a little more closely.

Using HTML, developers can run Java programs, called applets, over the Web. But Java is a completely different language from HTML, and it does not replace HTML. Java applets are automatically downloaded and executed by users as they surf the Web. The Java applet is run by the Web browser.

Another thing that makes Java special is that it was designed to be multi-platform. In theory, regardless of the machine and operating system that you are running, any Java program should be able to run. Many possible benefits accrue because Java enables developers to write an application once and then distribute it to be run on any

platform. These benefits include reduced development and maintenance costs, lower systems management costs, and more flexible hardware and software configurations.

So, to summarize, the major qualities of Java are:

- Its similarity to other popular languages.
- Its ability to enable Web interaction.
- Its ability to enable executable Web content.
- Its ability to run on multiple platforms.

Now that DB2 for OS/390 supports application development using Java, all of these qualities are available to DB2 applications.

JAVA BYTECODES AND JAVA VIRTUAL MACHINE (JVM)

After the Java program is written the program source code is compiled into machine-independent constructs called *bytecodes* using the Java compiler. Bytecodes are the manner in which Java achieves its platform independence. Because the Java bytecode is in a machine-independent, architecture-neutral format, it can run on any system with a standard Java implementation.

The Java bytecodes are then processed by the Java Virtual Machine (JVM). The JVM interprets the bytecodes for the platform on which the Java program is to be run. The JVM loads and verifies the Java bytecode. It is then passed to the Java interpreter to be executed. Alternately, the bytecodes can be passed to a Just-In-Time (JIT) compiler to be compiled into machine code to be executed.

Java has a reputation as a 'slow' language. That is, the performance of Java is questionable. The major disadvantage is that Java is an interpreted language, like BASIC. Both the Java interpreter and the JIT compiler consume resources and take time to process the Java bytecodes before execution. Therefore, the performance of a Java program will lag when compared to an equivalent program that is compiled and link-edited into object code (such as a C or COBOL program). As a developer, you must decide whether the platform independence and Web development capabilities offset the potential performance degradation.

JAVA APPLICATIONS, APPLETS, AND SERVLETS

There are three types of Java implementation method that you can use when accessing DB2 data from Java:

- Java applications
- Java applets
- Java servlets.

A *Java application* program is basically the same as a program written in any other programming language. It can perform all the tasks normally associated with programs, including many tasks that Java applets cannot perform. Furthermore, a Java application does not need a browser to be executed. It can be executed in a client or server machine.

A *Java applet* is a small application program that must be downloaded before it is run within a Java-enabled Web browser. Java applets reside on a Web server. When the Web server returns an HTML page that points to a Java applet, the Java-enabled Web browser requests the applet to be downloaded from the Web server. After the applet is received at the browser, either the browser starts the applet internally or an external JVM executes it.

Applets typically perform simple operations, such as editing input data, control screen interaction, and other client functionality. Of course, Java applets can be written to perform more complex functionality, but to load and run non-Java code in the client requires signed applets, which have the authority needed to run code in the client machine.

You should be aware that performance problems could arise because Java applets must be downloaded before they can be run. The time to download the applet must be factored into its service levels. In general, Java applets are small, so the performance impact should be negligible. But even small downloads can be slow if there are network problems. Java applets can be cached by the Web browser, which can diminish the performance impact.

A *Java servlet* is basically server-side Java. A Java servlet runs on the Web server, just like an applet runs in the Web browser. Java servlets

can be used to extend the functionality of the Web server. The Web server hands requests to the servlet, and the servlet replies to them. Servlets can be used instead of CGI applications.

To run Java servlets, your Web server must support the Java servlet API, developed by JavaSoft. This API defines how the servlet communicates with the server.

Java servlets have security advantages over client-side Java applets. A servlet that runs on a Web server inside a firewall can control access to sensitive data and business logic. Java applets do not inherently provide these security capabilities.

Before choosing which Java development style to use, you must know the basics of the environment in which the program must run. Ask the following questions when deciding what type of Java program is required for your development needs:

- How will the program be executed? Must it run over the Internet, as an intranet or extranet application, or merely as a stand-alone application?
- What is the business logic that this program must perform?
- How complicated is the program?
- How large (or small) is the program, and can it be quickly downloaded?
- What are the security requirements?
- Who are the target users and at what speed will they be connected to the Web?

Java applications, Java applets, and Java servlets are similar in nature. However, a different method is used to invoke each of them. Java applets and servlets are started from an HTML page. Java applications do not require a Web component, but can be used as part of an intranet solution.

To implement Java programs, you need to use the Java Developers Kit or JDK for short. The JDK is a development environment for writing Java programs conforming to the Java 1.1 Core API (Java 1.1 and later because that is the first JDK that included Java driver manager). The

JDK includes the Java Virtual Machine (JVM), Java classes, source files to create the classes in the JVM, documentation, and the JDK tools required for building and testing Java bytecode. These tools include the Java compiler and interpreter, the Java applet viewer, and the Java debugger.

JDBC AND SQLJ

Access to DB2 for OS/390 data in Java applications is accomplished using one of two methods – JDBC or SQLJ.

Java Database Connectivity (JDBC) is an API that enables Java to access relational databases. Similar to ODBC, JDBC consists of a set of classes and interfaces that can be used to access relational data. Anyone familiar with application programming and ODBC (or any call-level interface) can get up and running with JDBC quickly. JDBC uses dynamic SQL to access DB2 data. The primary benefits of JDBC include:

- The ability to develop an application once and execute it anywhere.
- JDBC enables the user to change between drivers and access a variety of databases without recoding their Java program.
- JDBC applications do not require precompiles or binds.

Potential drawbacks of JDBC include:

- JDBC uses dynamic SQL, which can add overhead when the SQL is bound.
- Programs using JDBC can become quite large.

The other way to access DB2 data from a Java program is SQLJ. SQLJ enables developers to embed SQL statements in Java programs. SQLJ provides static SQL support to Java. Developers can embed SQL statements into Java, and a precompiler is used to translate SQL into Java code. Then the Java program can be compiled into bytecodes, and a bind can be run to create a package for the SQL. Simply stated, SQLJ enables Java programs to be developed the way most DB2 programs have been developed for years.

Of course, SQLJ does not allow dynamic SQL. But you can mix SQLJ and JDBC in a single Java program, which effectively enables you to choose static or dynamic SQL for your Java programs. The primary benefits of SQLJ include:

- The ability to code static embedded SQL in Java programs.
- SQLJ source programs usually are smaller than equivalent JDBC programs.
- SQLJ does data type checking during the program preparation process and enforces strong typing between table columns and Java host expressions. JDBC passes values without compile-time data type checking.

Potential drawbacks of the SQLJ approach include:

- SQLJ programs must be precompiled and bound. So you will have to create and follow DB2 program preparation procedures for SQLJ Java programs.
- SQLJ is not yet a universally-accepted standard. However, it has been proposed to ANSI for inclusion in the SQL standard and it has the widespread support of the major RDBMS vendors.

Be sure to check out the Java sections of IBM's Web site for additional information regarding Java support and sample Java code. Two good URLs to bookmark are <http://www.ibm.com/developer/java/> and <http://www.ibm.com/software/data/db2/java/>.

USING RESULT SET ITERATORS TO RETRIEVE MULTIPLE ROWS

Traditional DB2 application programs written in host languages use a DB2 cursor to retrieve individual rows from a multi-row result set. The SQLJ equivalent of a cursor is a result set iterator. A result set iterator can be passed as a parameter to a method.

The result set iterator is defined using an iterator declaration clause specifying a list of Java data types. The Java data types represent columns of the table in the result set. The information in Figure 1 shows the SQL data types and their equivalent SQLJ data types that can be specified in result set iterator declarations. The SQLJ data type

<i>SQLJ data type</i>	<i>SQL data type</i>
java.lang.String	CHAR VARCHAR LONG VARCHAR GRAPHIC VARGRAPHIC LONG VARGRAPHIC
java.math.BigDecimal	NUMERIC INTEGER DECIMAL SMALLINT FLOAT REAL DOUBLE
Boolean	INTEGER SMALLINT
Integer	INTEGER SMALLINT DECIMAL NUMERIC FLOAT DOUBLE
Float	INTEGER SMALLINT DECIMAL NUMERIC FLOAT DOUBLE
Double	INTEGER SMALLINT DECIMAL NUMERIC FLOAT DOUBLE
byte[]	CHARACTER VARCHAR LONG VARCHAR GRAPHIC VARGRAPHIC LONG VARGRAPHIC
java.sql.Date	DATE
java.sql.Time	TIME
java.sql.Timestamp	TIMESTAMP

Figure 1: SQLJ and SQL data type equivalents

in the left column can be used for retrieved data that is of any of the SQL data types listed in the right column.

Note:

- The byte[] SQLJ data type is equivalent to the SQL data type with a subtype of FOR BIT DATA.
- The java.sql.Date, java.sql.Time, and java.sql.Timestamp data types are part of the JDBC API.

DBA CONSIDERATIONS

Before your shop moves forward with Java development under DB2 for OS/390, be sure that your DBAs have been trained in Java. Intelligent organizations require that the database administration group reviews all application code that accesses DB2 databases before the code can be promoted to production. If your DBAs do not understand Java, how will they be able to provide tuning advice for Java programs? The short answer is, they won't. So all DBAs should have at least rudimentary knowledge of Java constructs and the way in which Java programs are developed. Failure to educate the DBAs in the basics of Java is a sure-fire method for ensuring performance problems with DB2-based Java applications.

SUMMARY

The Internet is infiltrating every aspect of information technology. And DB2 is most definitely impacted. One of those impacts is the increased usage of Java.

Java is increasing in popularity not just because of its Web abilities, but also because of its portability. Now that DB2 for OS/390 supports Java, your organization may be the next one to begin using Java. Make sure that you are prepared to support Java and understand its benefits and drawbacks, and how it works, prior to using Java for production applications.

Craig S Mullins
Director, DB2 Technology Planning
BMC Software (USA)

© Craig S Mullins 2000

All those joins

Fuzzy SELECT (see *Fuzzy Select, DB2 Update, Issue 87, January 2000* for a description) usually requires joins to satisfy fuzzy queries. Join was simple in the olden days, ie:

```
SELECT    a, b, c, x, y, z    --a b c in table1; x y z in table2
FROM      table1, table2     --two or more tables to form the join
WHERE     a = x              --equijoin, unequal joins allowed
```

DB2 returns a result table containing a b c from table1 and x y z from table 2 whenever a = x. SQL programmers complained that the join was inadequate for certain queries such as those needing matching and non-matching rows (outer joins). Assume the following simple logical tables and values shown in Figures 1 and 2.

ALUMNUS(<u>ALUMNUS_ID</u> , surname, EMPLOYER_ID)		
alumnus_id	surname	employer_id
24	gluck	22
31	monroe	-
34	latimer	62
43	pabst	-
61	finn	32
<i>Figure 1: Alumnus table values</i>		
EMPLOYER(<u>EMPLOYER_ID</u> , business_name)		
employer_id	business_name	
22	ibm	
26	ingres	
32	aurora	
55	topsys	
62	hp	
<i>Figure 2: Employer table values</i>		

Assume you want a complete alumni list with their employer information. The following join seems sufficient:

```

SELECT      alumnus_id, surname, a.employer_id, business_name
FROM        alumnus a, employer e
WHERE       a.employer_id = e.employer_id
ORDER BY   alumnus_id

```

The SQL above returns the values in Figure 3.

alumnus_id	surname	employer_id	business_name
24	gluck	22	ibm
34	latimer	62	hp
61	finn	32	aurora

Figure 3: Standard join

It does not work because some alumni are retired or unemployed. DB2 skips them since there is no matching row in employer table. One solution (there are others) is:

```

SELECT      alumnus_id, surname, a.employer_id, business_name
FROM        alumnus a, employer e
WHERE       a.employer_id = e.employer_id
UNION
SELECT      alumnus_id, employer_id
FROM        alumnus
WHERE       NOT IN
            (SELECT      employer_id
             FROM        employer)
ORDER BY   1          --requires column number

```

The first SELECT returns Figure 3. The second SELECT returns 22, 26, 32, 55, and 62 for the NOT IN clause. The result table for the second SELECT is shown in Figure 4.

alumnus_id	surname	employer_id
31	monroe	-
43	pabst	-

Figure 4: Intermediate result table for second SELECT

DB2 combines the two intermediate result tables, eliminating duplicate rows (none exist) returning Figure 5.

alumnus_id	surname	employer_id	business_name
24	gluck	22	ibm
31	monroe	-	-
34	latimer	62	hp
43	pabst	-	-
61	finn	32	aurora

Figure 5: Outer join simulation using join, UNION, and subquery

DB2 now has many joins:

```

>>table-reference | INNER      |-----|
|-----|-----| JOIN  table-reference ON
|-----|-----| join-condition      ><
|-----| OUTER      |
| LEFT--  |
| RIGHT-- |
| FULL--  |
>>(joined-table)----- ><
[(joined-table) specifies an intermediate result table by applying one
of the join-operators.]

```

An INNER JOIN (default) is the equivalent to a standard join producing the same result table as Figure 3:

```

SELECT      alumnus_id, surname, a.employer_id, business_name
FROM        alumnus a
INNER JOIN  employer e
ON          a.employer_id = e.employer_id
ORDER BY   alumnus_id

```

Figure 4 is a LEFT OUTER JOIN since it keeps all rows from the left or first table:

```

SELECT      alumnus_id, surname, a.employer_id, business_name
FROM        alumnus a
LEFT OUTER JOIN  employer e
ON          a.employer_id = e.employer_id
ORDER BY   alumnus_id

```

RIGHT OUTER JOIN keeps all rows from the right or last table:

```

SELECT      alumnus_id, surname, a.employer_id, business_name
FROM        alumnus a

```

```

RIGHT OUTER JOIN  employer e
ON                a.employer_id = e.employer_id
ORDER BY         e.employer_id

```

Implicit INNER JOIN precedes RIGHT OUTER JOIN. The INNER JOIN above returns Figure 3 as before. A RIGHT OUTER JOIN returns Figure 2. Combining Figure 3 and Figure 2 returns Figure 6:

alumnus_id	surname	employer_id	business_name
24	gluck	22	ibm
-	-	26	ingres
61	finn	32	aurora
-	-	55	topsys
34	latimer	62	hp

Figure 6: Combined result table for RIGHT OUTER JOIN

A FULL OUTER JOIN keeps all the rows of both tables:

```

SELECT          alumnus_id, surname, a.employer_id, business_name
FROM            alumnus a
FULL OUTER JOIN employer e
ON              a.employer_id = e.employer_id
ORDER BY       a.employer_id

```

An *Implicit* INNER JOIN returns Figure 3 whereas A FULL OUTER JOIN returns Figure 1 *and* Figure 2. Combining the three tables returns Figure 7.

alumnus_id	surname	employer_id	business_name
24	gluck	22	ibm
31	monroe	-	-
34	latimer	62	hp
43	pabst	-	-
61	finn	32	aurora
-	-	26	ingres
-	-	55	topsys

Figure 7: Combined result table for FULL OUTER JOIN

Life can get complicated if you use additional predicates or WHERE:

```

SELECT      alumnus_id, surname, a.employer_id, business_name
FROM        alumnus a
INNER JOIN  employer e
ON          a.employer_id = e.employer_id
AND         business_name = 'ibm'
ORDER BY   alumnus_id

```

The SQL above returns 24/gluck/22/ibm, but may have unanticipated results if you substitute FULL OUTER JOIN:

```

SELECT      alumnus_id, surname, a.employer_id, business_name
FROM        alumnus a
FULL OUTER JOIN  employer e
ON          a.employer_id = e.employer_id
AND         business_name = 'ibm'
ORDER BY   alumnus_id

```

The implicit INNER JOIN intermediate result table is shown in Figure 8.

alumnus_id	surname	employer_id	business_name
24	gluck	22	ibm

Figure 8: Intermediate result table for additional predicate

A FULL OUTER JOIN again returns Figure 1 and Figure 2. Combining the three tables produces Figure 9.

alumnus_id	surname	employer_id	business_name
24	gluck	22	ibm
31	monroe	-	-
34	latimer	62	hp
43	pabst	-	-
61	finn	32	aurora
-	-	26	ingress
-	-	32	aurora
-	-	55	topsys
-	-	62	hp

Figure 9: FULL OUTER JOIN with additional predicate

Inspection of Figures 7 and 9 reveals that Figure 9 has two more rows, although both are FULL OUTER JOINs of the same tables. The

additional predicate returns one row (Figure 8) therefore DB2 must place all the other rows in the combined result table. Figures 1 and 2 each have five rows so $(2 \times 5) - 1$ [duplicate row] = 9 rows instead of 7.

You must realize that you will get additional rows in a combined result table when you use additional predicates and those rows must be anticipated.

WHERE is applied *after* the combined intermediate result table has been created:

```
SELECT      alumnus_id, surname, a.employer_id, business_name
FROM        alumnus a
FULL OUTER JOIN  employer e
ON          a.employer_id = e.employer_id
WHERE       business_name = 'ibm'
```

The SQL above returns Figure 10 because 24/gluck/22/ibm is the only row that satisfies the WHERE condition.

alumnus_id	surname	employer_id	business_name
24	gluck	22	ibm

Figure 10: Combined result table using WHERE

Here is another example to illustrate the very different results achieved by using an additional predicate or WHERE. Assume that business_name is changed to ingres:

```
SELECT      alumnus_id, surname, a.employer_id, business_name
FROM        alumnus a
FULL OUTER JOIN  employer e
ON          a.employer_id = e.employer_id
AND         business_name = 'ingres'
```

The SQL above returns Figure 11 because the INNER JOIN intermediate result table is empty requiring DB2 to combine Figure 1 and Figure 2.

Using WHERE:

```
SELECT      alumnus_id, surname, a.employer_id, business_name
FROM        alumnus a
```


alumnus_id	surname	employer_id	business_name
24	gluck	-	-
31	monroe	-	-
34	latimer	-	-
43	pabst	-	-
61	finn	-	-
-	-	22	ibm
-	-	26	ingres
-	-	32	aurora
-	-	55	topsys
-	-	62	hp

Figure 11: Combined result table when INNER JOIN returns no rows

```

FULL OUTER JOIN  employer e
ON               a.employer_id = e.employer_id
WHERE           business_name = 'ingres'

```

The SQL above returns Figure 12.

alumnus_id	surname	employer_id	business_name
-	-	26	ingres

Figure 12: Combined result table using WHERE

As before, INNER JOIN returns no rows so FULL OUTER JOIN returns Figure 11. WHERE is applied leaving Figure 12. Be careful when using additional predicates or WHERE!

Fuzzy SELECT sometimes needs relational division. Relational division of two tables returns rows where column values in the first table match column values for every corresponding value in the second table. DB2 does not have a join-operator so a simulation must be used. My tables are too simple so I will use the DB2 sample tables of DSN8230.PROJECT and DSN8230.ACT.

```

SELECT  projno
FROM    dsn8230.project  pjact

```

```

WHERE      NOT EXISTS
           (SELECT    actno
            FROM      dsn8230.act    act
            WHERE     NOT EXISTS
                 (SELECT    projno
                  FROM      dsn8230.project    pjact2
                  WHERE     pjact.projno = pjact2.projno
                  AND       act.actno = pjact2.actno)

```

The SQL above uses three correlated subqueries to return all projects requiring every activity listing in the activity table.

Eric Garrigue Vesely
Principal/Analyst
Workbench Consulting (Malaysia)

© Xephon 2000

Making sure that Business Intelligence aids intelligent business with DB2 Query Patroller – part 2

DB2 QUERY PATROLLER TRACKER

The DB2 Query Patroller Tracker tool enables administrators to manage databases by displaying usage history in a graphical user-friendly format. It provides two key features that support system administrators in managing a DSS. Firstly, it gives them the ability to monitor the database load and activity over time. Secondly, it provides the administrator with details of table and column accesses to assist in tuning the system. In addition to the reports that Query Patroller Tracker provides, the Query Patroller server stores all of its historical information in DB2 tables so that administrators can drill down on whatever aspects of the database usage they desire, using the query tool of their choice. The functions of Query Patroller Tracker are valuable to charge-back costings to departments using that DSS and may lead to better decisions with respect to access schedules, selection of courses for education, or resource limitations.

Figure 1 shows an example of the Query Patroller Tracker.

The screenshot shows the DB2 Query Patroller Tracker interface. On the left is a tree view of database objects under the 'db2test' schema, including ACCT, CUBE_AST, CUST, PG_SALESSUM, PROCLINE, SPSS_CUBE, SYSTABLES, TRANS, TRANSITEM, and Jobs. On the right is a table titled 'Total Table Hits' with the following data:

Table	Number of hits	Percentage of hits	Number of rows returned	Percentage of rows returned
SYSS_CUBE	76	55.06	2	0.3
CUBE_AST	36	26.47	8	8
PG_SALESSUM	9	6.62	8	8
CUST	4	2.94	8	8
PROCLINE	4	2.94	2	0.8
TRANS	3	2.21	8	8
ACCT	2	1.47	8	8
SYSTABLES	1	0.74	218	98.2
TRANSITEM	1	0.74	8	8

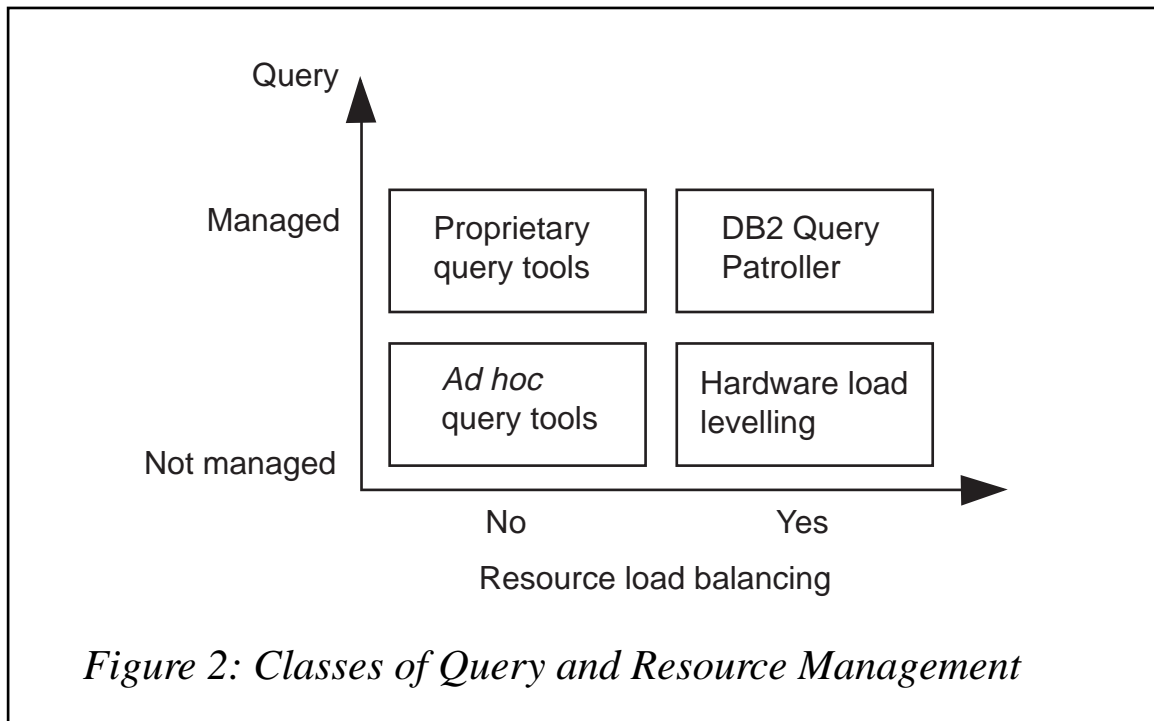
Figure 1: DB2 Query Patroller Tracker

DB2 QUERY PATROLLER BRINGS IT ALL TOGETHER IN ONE PRODUCT

To understand DB2 Query Patroller functionality and how it differs from query tool management systems, it is necessary to understand the problems each tool is designed to solve. At present, four technologies are available to manage queries and resources:

- *Ad hoc* query tools
- Three-tier proprietary tools
- Server-based query and resource managers
- Hardware resource managers.

Each has its own strengths that make it appropriate for particular types of query situations. Figure 2 divides queries into four classes, across two axes, based on the resource load levelling provided and the management of that query before it runs against the database.



AD HOC QUERY TOOLS

Ad hoc query tools do a good job of allowing end users to directly ask questions of the DSS without having to go to IS personnel every time they need additional information. When databases are small and the number of users is low, there is little need for query and resource management. However, as databases grow larger and the number of users increases, the strain on the DSS becomes evident. In some cases, the tool may have a rudimentary scheduling facility that requires users to keep their PCs powered on overnight to schedule the query or puts the burden on the end user to share resources in good faith with other users.

HARDWARE LOAD LEVELLERS

Some database management systems and MPP hardware companies offer system software products that spread the query load across the database partition servers. Queries are routed to free database partition servers for processing. Even though this provides a good use of the hardware resources, it does not look at the type of query being submitted. Any query that comes into this type of system is immediately

run, regardless of the time or cost that query will consume. This type of tool is at the mercy of those runaway queries.

PROPRIETARY QUERY MANAGEMENT TOOLS

Three-tier query tools have a server component that provides some capabilities for scheduling queries. This component releases the desktop and submits the query at the scheduled time on behalf of the end user. Typically, these types of tool work only with their own front end and provide a pre-canned query interface for end users. This is less adaptable for *ad hoc* querying and more for managing the query environment. Typically, many users submit very predictable optimized queries. Three-tier query tools provide little user prioritization and resource balancing.

DB2 QUERY PATROLLER – ALL MANAGEMENT IN ONE TOOL

The first three categories of query and resource management tools fail to provide end users with guaranteed query response times with well written queries, nor do they provide IS with the control they need. DB2 Query Patroller addresses these challenges. DB2 Query Patroller is the only DB2 product that controls and monitors queries. DB2 Query Patroller works with dynamic SQL query tools to prioritize and schedule user queries based on user profiles and cost analysis performed on each query. Large queries are put on hold and can then be scheduled for a later time during off-peak hours so that they do not significantly impact other end users during regular business hours. Queries with a high priority (based on user profiles) are promoted to the top of the schedule. In addition, DB2 Query Patroller monitors resource utilization statistics to determine which database partition servers are the least busy and provides load distribution functionality to even out the workload across the system.

HIGH AVAILABILITY, STABILITY, AND ROBUST PERFORMANCE

DB2 Query Patroller provides a highly available and stable environment for *ad hoc* query analysis against a DSS.

PROACTIVE QUERY CAPTURE

At the core of DB2 Query Patroller's breakthrough functionality is its ability to proactively capture queries. Query Patroller's proactive approach to its queries helps it guarantee the high levels of availability and stability required. As queries are submitted against the DSS, Query Patroller traps the queries, assesses their cost, and prioritizes their execution. Without this proactive query trap, users would submit queries and you could only report in retrospect why the system failed. Notice the word failed? It is past tense. The point is that you would have to wait three or four hours (or even days) before you realized the query was a runaway query. The cost of the resource is hard to measure, but surely has an impact with respect to user's productivity and the costs associated with DSS management. DB2 Query Patroller serves as a vigilant eye over the vital corporate DSS and allows you to leverage what was once wasted resource. Also note that since the queries are captured, should the database server ever fail, these queries will be automatically restarted by DB2 Query Patroller on behalf of the end user.

ROBUST QUERY TERMINATION

This proactive approach is enhanced through DB2 Query Patroller's robust query termination. One of its strengths is its ability to effectively kill queries. Many *ad hoc* query tools give end users the capability to terminate a query, but in reality the query is just terminated on the client workstation. The processes already started on the database server are not notified and continue to run. If the user assumes that a query has been cancelled, they might be more likely to submit other queries and repeat the same submit and terminate process. The end result? The DSS gets bogged down with multiple orphan queries that continue to run, wasting valuable resources.

DB2 Query Patroller addresses this problem by sending a force command to the database. This command ensures that both the end user workstation and the database server are released from processing a terminated query. Multiple benefits are obtained with the implementation of this solution. Firstly, it ensures that the cycles used for processing on the database server are fully utilized by needed queries, and secondly it frees up the system administrator from having to monitor and kill the orphan queries.

FREES USER DESKTOP, IMPROVING PRODUCTIVITY

Typically with most front-end query tools, after a user submits a query, the user's application freezes in a 'pending output' state until the results of the query return. Users must wait for their desktops to become available when the query completes, which can greatly reduce their productivity. Being on hold greatly increases the likelihood that an end user will terminate a running query in order to regain control of the workstation. Users also need to be able to perform other tasks, even submit additional queries, while earlier queries run in the background.

In many cases, users don't need their query results back until the next day or the following Monday morning. Thus, instead of submitting a query immediately, the query could be scheduled to run at a later time when the system load may be lower. DB2 Query Patroller frees up the user's application and improves productivity by allowing users to submit and schedule queries based on their response requirements.

QUERY COST ANALYSIS

DB2 Query Patroller is integrated with the DB2 optimizer. The DB2 optimizer performs a cost analysis for each query submitted against the system and determines the static cost of the query. In DB2, the cost of a query is measured in timerons. DB2 Query Patroller enables the system administrator to modify a user's profile and specify a query cost threshold for each user or group. After completing cost analysis, DB2 Query Patroller compares the returned value to the value in the user profile. If the returned value exceeds the user threshold, DB2 Query Patroller places the query on hold so that the query can run at a later time or be terminated. This is a key feature in preventing runaway queries from inexperienced users reaching the DSS.

USER PRIORITIZATION

The majority of *ad hoc* query tools do not take into account a user's priority with respect to other users submitting queries against the DSS. For example, often the CEO of a company needs a report right away for a meeting, but the system is so overloaded that the query does

not complete in time. Because of the CEO's priority level, the query request should automatically move to the top of the query queue and be executed immediately.

DB2 Query Patroller provides an environment that facilitates the timely completion of queries. It maintains a user profile for each user who submits queries into the system. The user profile defines a priority class, which identifies the relative priority a user has when submitting a query against a DSS. A higher priority class places the user's query closer to the top of the query submission queue.

The system administrator sets individual user and group priorities, thus enabling the DSS to deliver the needed results to the most important users as quickly as possible.

DB2 Query Patroller also enables the system administrator to limit the number of queries that each individual user can simultaneously submit. This feature gives other users the opportunity to have their queries processed in a timely fashion.

LOAD BALANCING

After a query leaves the queue, it should be submitted to an area of the hardware resource that is being used the least. For example, in an MPP environment, *ad hoc* query tools may submit queries to one or two database partition servers on the system. Then, the database management system takes the query and runs it on the appropriate partitions. In comparison, a server-based query manager could decide which database partition server the query should be submitted to and could send it there immediately. This elimination of one extra step in the process prevents bottlenecks if the database partition servers are bombarded with *ad hoc* query requests.

DB2 Query Patroller provides a system administrator with the ability to set system and user parameters to govern the queries entering the DSS. A system administrator may specify the maximum number of concurrent queries for each user, for each database partition server, and for the entire system. DB2 Query Patroller provides load levelling across MPP hardware environments. By tracking database partition

server utilization, Query Patroller routes queries to idle database partition servers and spreads the query load across the system.

DATA WAREHOUSE OPTIMIZATION

DB2 Query Patroller enables end users and system administrators to monitor the database load by providing access to the following information:

- Tables being accessed for all jobs
- Columns accessed for each table
- Number of rows returned, by table, for all jobs
- Detailed view of job activity over time
- Historical view of job activity.

DB2 Tracker displays this information in an easy-to-view format by determining the total number of tables accessed in the database and calculating the total number of times each specific table is accessed.

For each table displayed, the user is able to drill down to view the columns accessed for queries against that table. This enables the administrator to decide whether new indexes should be created on the columns queried the most in the table.

DB2 Query Patroller is an indispensable tool for any DSS. End users at all levels can enjoy access to the data that leads to more profitable and accurate decisions, while, at the same time, IS can guarantee that the DSS will be available to senior management.

If you are interested in learning more about DB2 Query Patroller and the other products in the DB2 UDB database family, please contact your local IBM representative or visit the IBM Web site at: <http://www.software.ibm.com/data/db2>.

IBM and DB2 are trademarks or registered trademarks of International Business Machines Corporation in the United States of America and other countries.

*Paul Zikopoulos
Database Specialist with the DB2 Sales Support team
IBM Canada (Canada)*

© IBM 2000

DB2 Versions 6.2 and 7

It has come a long way; DB2 is now about to flower into adulthood – and the smart money is saying that IBM has listened carefully to its users before deciding what to include in the new Version 7 implementation due for release just before Christmas this year.

Big Blue released what it called its maintenance version of Version 6 – popularly known as V6.2 – in March. While V6 had some heavyweight upgrades in it, V7 includes all the tweaks that the IBM user community has been asking for over the past couple of years.

IBM accepts that a lot of sites have not wanted to move from V4 and V5 until they were really satisfied that all their Y2K compliance work really was stable. However, V7 is the upgrade everyone has been waiting for – so sites need to start planning now to ensure trouble-free upgrades.

What does the IBM Computer Users Association think of V6.2 and V7? Spokesman Geoff Petherick noted: “Our members generally find that DB2 is much better at handling large jobs compared to (say) Oracle – although Oracle Corp is currently much better at marketing its flagship relational database than Big Blue is.

“Oracle’s packaging looks better and can easily give you a warm feeling. The major difference is that IBM, in association with its business partners and training companies, gives you vastly more tuning information. You can really play with the knobs on DB2 to boost throughput.

“Sites should not under-estimate the importance of being able to tune their relational databases. Y2K compliance programmes have also shown the nation’s boardrooms how much hardware and licence fee savings there are to be had from well-tuned systems”, he added.

So why do sites need to prepare for V6.2 and V7 now? Petherick, who is a director of both GSoft and RS Software, replied: “The e-commerce revolution is here to stay.

“This means that sites have not only got to be able to deliver 24 x 7 service, but do it as efficiently as possible. There are plenty of end users who feel that responses to queries for jobs like data mining are not coming in fast enough.

“The growth of Service Level Agreements (SLAs) and charging out IT to end user departments also means that sites will need much more accurate costing information on how much resource the queries really cost to run.

“So sites need to plan and train for V6.2 and V7 now. To gain the maximum benefit from V7 (and hence the maximum savings in resources) developers may need to change their approach to DB2 implementation and support.

“Don’t forget that the unofficial motto of the SAS is ‘Who trains, wins’”, he added.

Although IBM has offered costing facilities inside DB2 for some time, these have tended to cover CPU utilization only. Petherick said he was pleased to see that IMSI’s Query Costing Facility (QCF) was set for release in V7.

IMSI’s chief executive is Frank Ingrassia, who observed: “QCF can be retroactively fitted to V5 and V6. The aim of my software is to allow sites to get very accurate costings on how much queries really cost to run. DB2 has its own facilities which have been good at the CPU level.

“QCF takes into account all the resources so that developers can utilize different techniques for running queries (eg trying to find the shortest path). It supports DB2, TSO, and batch jobs. There is a stored procedure version for IMS, CICS, etc.

“This allows end users to govern their own budgets better because they will be able to see how much queries really cost to run. I believe this will lead to developers being trained to write better queries and applications in a DB2 environment”, he added.

So what can techies expect to find in V6.2? Ingrassia replied: “The major upgrades in V6.1 and 6.2 show that IBM is aggressively

pursuing the DB2 utilities market. This includes upgrading the database's tool set so that it is as good as (or superior to) third-party offerings previously only available from BMC Software (Online Zparm Changes) or Platinum/CA (Fast Unload).

“Combined with Icing Software, which offers tools like Buffer Pool Tool or SQL Performance Analyser, IBM is making good its threat to curtail software licensing costs as a percentage of ownership.

“This dovetails nicely with recent announcements on IBM's 64-bit mainframe technology, Freeway, and its new server partitioning and related costing capabilities”, he pointed out.

While V6.2 is a refresh for V6 with all APARs to date, there are a number of new features, some of which were made available as V5/6 APARs.

Ingrassia, who is an IBM Gold Consultant, said: “Developers need to check carefully which can be added to V5 and which will be available only once the site has upgraded to V6.2 or V7.

“New features include the DB2 Control Centre for the universal database for OS/390, SQL upgrades (which has been retrofitted to V5) covering BEGIN . . . END, local variable declarations, assignments, CASE and IF statements, as well as WHILE, REPEAT, FOR, LEAVE, and RETURN statements in loops.

“The Stored Procedure Builder has also been retrofitted to V5 and there are upgrades to the REXX language, SQL, and stored procedures. This also applies to Java (SQLJ and Java stored procedures).

“There are enhancements to the star join optimization and 15 table joins (this is equal to 15 squared to 225 tables maximum).

“DECLARED TEMPORARY tables can be defined on-the-fly. The instruction DECLARE GLOBAL TEMPORARY TABLE also supports full INSERT, UPDATE, and DELETE.

“External save points allow developers to roll back a portion of their unit of work, issue SQL, SAVEPOINT ABC, issue more SQL, and then ROLLBACK TO SAVEPOINT ABC.

“In identity columns, DB2 V6.2 automatically generates unique values for a column. For example, CREATE TABLE MYTABLE (COL1 INTEGER . . . COL7 INTEGER) GENERATED ALWAYS AS IDENTITY (START WITH 10 INCREMENT BY 1).

“The DEFER DEFINE avoids having to use resources to generate empty tables and index spaces. The table or index is generated only when data is inserted.

“You can use SELECT as a source of an UPDATE SET statement and use UPDATE with SUB-SELECT. The IFI consolidation feature allows for easier data sharing problem resolution”, he added.

IBM has tried to be a lot more open about what is available under V7 than has been the case in the past. It has produced two booklets, one called *DB2 Universal Database Server for OS/390 – What’s New V7* and is aimed at giving an executive overview, while the other is called *IBM DB2 Universal Database – What’s New V7* and is aimed at techies.

It is worth getting hold of both booklets so that sites can compare the differing messages for the suits and the techies. Both are commendably well-written and the executive version would also be helpful for any developers who are new to DB2 or are returning to support of this database after baby breaks, etc.

Although DB2 has been implemented on a number of other platforms apart from MVS, Big Blue recognizes that most of the work (and hence revenue) currently being done in the DB2 world is at the mainframe level.

Much of the research is done in the UK at IBM Hursley near Winchester.

IBM is keen to promote its flagship relational database on non-mainframe platforms and recent independently-observed benchmarking against two other Big Five RDBMSs confirmed that DB2 was able to show companies whose names begin with ‘O’ and ‘S’ a clean pair of heels.

Hursley claims the only difference between the various

implementations of DB2 is the way in which it relates to the underlying operating system and files. This is an area where IBM needs to do a lot of work in convincing sites that this really is the case.

According to Hursley's boffins, the main thrust of V7 is to give techies very high availability, quality tools, more usability, greater portability between the major ERP suites, as well as features to encourage sites to develop more resource-hungry business intelligence and data warehouse applications.

While this may sound like a licence to print lots more orders for the IBM sales force, Hursley believes that moving BI and DW applications off so-called 'low-cost platforms' such as Unix and NT will actually save sites more money as well as bandwidth. Most independent studies for the past five years have confirmed that mainframes are the best value for money for sites with large populations of users.

The IBM CUA believes that Big Blue has made an honest attempt to deliver the extra knobs that developers have been asking for and IMSI's Ingrassia believes V7 is the release worth waiting for.

He pointed out: "From the point of view of SQL usability, V7 includes scrollable cursors. This allows users to scroll forward and backward in their results sets. The `FETCH FIRST n ROWS ONLY` clause actually limits the answer set to n rows – unlike the `OPTIMIZE FOR n ROWS` clause. There are more optimizer improvements from Starburst as well as statistics history tables.

"The V7 optimizer technology will allow techies to correlate subqueries using indexes as Stage 1. In-lists will be processed in parallel and there are join enhancements to the star schema. On-line changes cover `ALTER, TABLE, COLUMN, and PARTITION`. There are also online Zparms with a few dozen enabled.

"V7 utility improvements cover `ONLINE LOAD RESUME, PARALLEL LOAD` (for multiple inputs) object wildcarding, `FAST UNLOAD` enhancements, and on-line reorganization without dataset rename and drain readers.

"Data sharing improvements cover group attach and restart enhancements, coupling facility lock propagation reduction, and

avoiding global drain for lock page lists.

“You can also update and delete with self-referencing SELECTs, execute multi-row fetches (FETCH FIRST *n* ROWS ONLY), and use UNION and UNION ALL in views.

“IBM is currently working on functionality beyond V7.1, which should be available next year. So watch out for NO LOG tablespaces, bi-directional index scans, recursive SQL, and summary tables as future features”, Ingrassia added.

Hursley has two managers responsible for the V7 launch: Sara Ellis is responsible just for the mainframe world while Paul Read has overall responsibility for all platforms the database runs on.

Read is keen to push the BI, DW, and Internet aspects of V7. He pointed out: “In addition to the BI and DW features, we are offering Data Warehouse Manager for MVS which has the same look and feel as DB2. We have included the Visual Warehouse option for the workstation implementation. This includes OS/2, Unix, and Windows.

“There is an OLAP Start-up Kit with three connections. This will allow sites to bundle functions like rolling averages and cubes.

“We see the XML enhancements as key to breaking into the B2B (business to business) and B2C (business to consumer) markets. This will allow developers to integrate their DB2 workloads more closely with Websphere and our Enterprise Information Portal (EIP).

“IBM wants to make DB2 tools more accessible, so we have added and improved SQL Performance Analyser, Recovery Manager, Bind Manager, and put extra facilities on the SQL stored procedures feature on our workstation versions.

“The XML extenders will allow developers to integrate better with net.data and the spatial extenders are aimed at sites that use and manage lots of spatial data such as in insurance.

“OLE, JDBC, and SQLJ are fully supported. This is the first phase of a federated database strategy so that applications can read from Oracle and SQL Server using our own optimizer technology.

“IBM acknowledges that DB2 has been perceived as a mainframe-

only database for years. During the past 24 months we have done a lot of work to change that perception into one of a universal database.

“We expect to pick up a lot of new business from sites that are just starting out with Linux or want to deploy DB2 on their departmental machines. They now know there will be no problems in scaling up their DB2 workloads onto larger machines.

“DB2 is the first relational database to be certified for use on Windows 2000 and it integrates both with Visual Studio and VisualAge. This gives developers and administrators DB2 icons on their screens”, Read added.

An area most relational database vendors are perceived to have overlooked is that of heritage databases. About 80% of the world’s data sits under the likes of IDMS, IMS, ISAM, and VSAM databases. Many sites expect IBM to provide better connections between the heritage and DB2 worlds.

Read replied: “We have the Connect family of products. Classic Connect goes into heritage file systems, there is Relational Connect for other RDBMSs, and DB2 Connect, which allows sites to get faster data transfer rates between different platforms running that database.”

What features in V7 did Hursley expect the majority of developers to want? Sara Ellis replied: “The first is on-line system parameter changes. On previous versions you would have to take down the system. There is also on-line reorganization, which will increase availability.

“We have also designed in more consistent restart facilities so that sites can re-load or re-organize the work themselves.

“Traditionally we had only an open cursor that moved forward. The scrollable cursor will allow developers to go forwards and backwards.

“SP sites have been very keen on the UNION and UNLOAD statements. Union works on two SELECTS or more, and in the control centre customers can now request an UNLOAD prior to DSNT. This will help with third-party software: for example, unloading from field selection and then formatting logical processing. This would allow you to unload every 50th row or just the first 1,000 rows (say).

“We have upgraded extender facilities, which cover data sharing enhancements and the restarting light option. For example, if a DB2 member in a group goes down, you can now bring up the subsystem on another subsystem. This frees up the locks. Previously you would have needed a lot of storage to cover that eventuality.

“We have enhanced utility tests and dynamic allocation features. For example, rather than having to specify individual tables, developers can use the wildcard facility to copy a complete image.

“DB2 dynamically allocates image copies, etc via templates rather than having to write the JCL. The database also keeps a history of statistics inside its catalog. These are regularly updated so that developers can see growth over a period of time. This will be useful for capacity planning.

“Sites can use the COMMIT verb and roll-backs through stored procedures. Currently developers are not able to combine these with save points. This was introduced in V6.

“As promised, IBM has put more functionality into the Java side. This covers stored procedures, which we expect to be very popular in workstation implementation (eg OS/2, Unix, and NT).

“A number of mainframe sites have expressed interest in the Java enhancements, but have been treading water up till now. Enhancements include upgrades to allow Java code to run in native mode inside the JVM (Java Virtual Machine) rather than being compiled”, Ellis concluded.

IBM Hursley has invited all readers to look at the home page on www.ibm.com/software/data/db2/OS390 for a full list of V7 enhancements.

SAP

For many developers the biggest curse of having to implement and maintain so-called ERP (Enterprise Resource Planning) applications is the vast numbers of tables – typically between 1,000 and 10,000 – which must be installed irrespective of whether the site will be using those facilities or not.

While this is very good for the sales of disk drives, IBM has been looking at ways of reducing the initial hardware overhead of ERP packages because all suppliers and their channels routinely roll out the implementations department by department.

So it is worth checking how IBM is helping to ease the lot of developers blessed with this Herculean task. The first has been to pre-configure mainframes to support the SAP R/3 environment. This is called the SAP Application Server and should make it easier for sites to plug into Lotus Domino, MQSeries, and VisualAge Java facilities.

The second is tucked away in V7. According to Hursley's Sara Ellis, the DEFER DEFINE facility has been implemented specifically to help ERP sites who are using DB2.

She said that the aim was not to have to physically create a table until the site actually had some data to insert. Obviously the table will be defined to comply with the requirements of the ERP system

WINDOWS

Although most DB2 work is done in a mainframe environment, following consultations with its user groups, IBM has made a number of enhancements at the workstation end for those sites that wish to deploy the flagship relational database in the Windows world.

These V7 enhancements are specifically designed for Windows 2000 developers with the aim of making DB2 look and feel as if it is part of the Microsoft empire.

DB2 services are published in the active directory containing the protocol configuration information that allows client applications to connect to a DB2 database server.

Authentication is achieved via the single sign-on facilities of Kerberos – but DB2 access is unchanged for sites who are not using the Microsoft product.

The DB2 Universal Database Control Centre can also be launched from the Microsoft Management Console, thus giving system administrators a simplified job (ie no sneakernet between different consoles).

Application development with Microsoft's controversial Visual C++ has been eased thanks to two components using embedded SQL. These are a tool bar (DB2 Visual C++ Tools Add-in) that launches administration and development tools from within the Visual C++ IDE (Integrated Development Environment). The second comprises management tools and wizards (DB2 Visual C++ Project Add-in) that help techies develop, package, and deploy client applications and stored procedures for DB2 servers running inside Windows 2000.

This only leaves the twin problems of Windows 2000 trying to grab extra hardware resources and the difficulties in developing portable and re-useable C++ unless the standards are rigidly adhered to.

Chris Youett
Freelance Journalist (UK)

© Xephon 2000

DRDA connectivity for DB2 UDB for OS/390

INTRODUCTION

DRDA stands for Distributed Relational Database Architecture. It is a set of architectural rules announced by IBM in June 1990 and since then it has been enhanced significantly by IBM.

DRDA allows exchange of computer data among relational DBMSs across the enterprise that resides on heterogeneous and unlike hardware platforms and operating systems.

In DRDA, the participating DBMSs are classified as either servers or requesters of data.

Most vendors have accepted DRDA, and, in October 1998, DRDA was endorsed officially by the Open Group Organization. Even Oracle, the archrival of IBM, expressed intent to comply with DRDA rules if its customer base demands it.

DRDA significance has been enhanced by the advent of Enterprise Resource Planning (ERP) applications and most recently by Web enabling technology, E-commerce, and CRM applications.

This article will list various DRDA connectivity configurations from a workstation to DB2 OS/390. The focus will be to detail the role of various DRDA products such as DB2 Connect, the Communication Manager, VTAM, and relevant DB2 OS/390 connectivity parameters (including usage of TCP/IP and SNA protocols).

Some common errors in connectivity that may be encountered in the above configurations will be discussed along with suggested solutions to these common errors.

Because of the detailed nature of this article it could be considered as a tutorial.

CONNECTIVITY CONFIGURATION OPTIONS

Direct 2-tier connectivity from a workstation to UDB DB2 OS/390

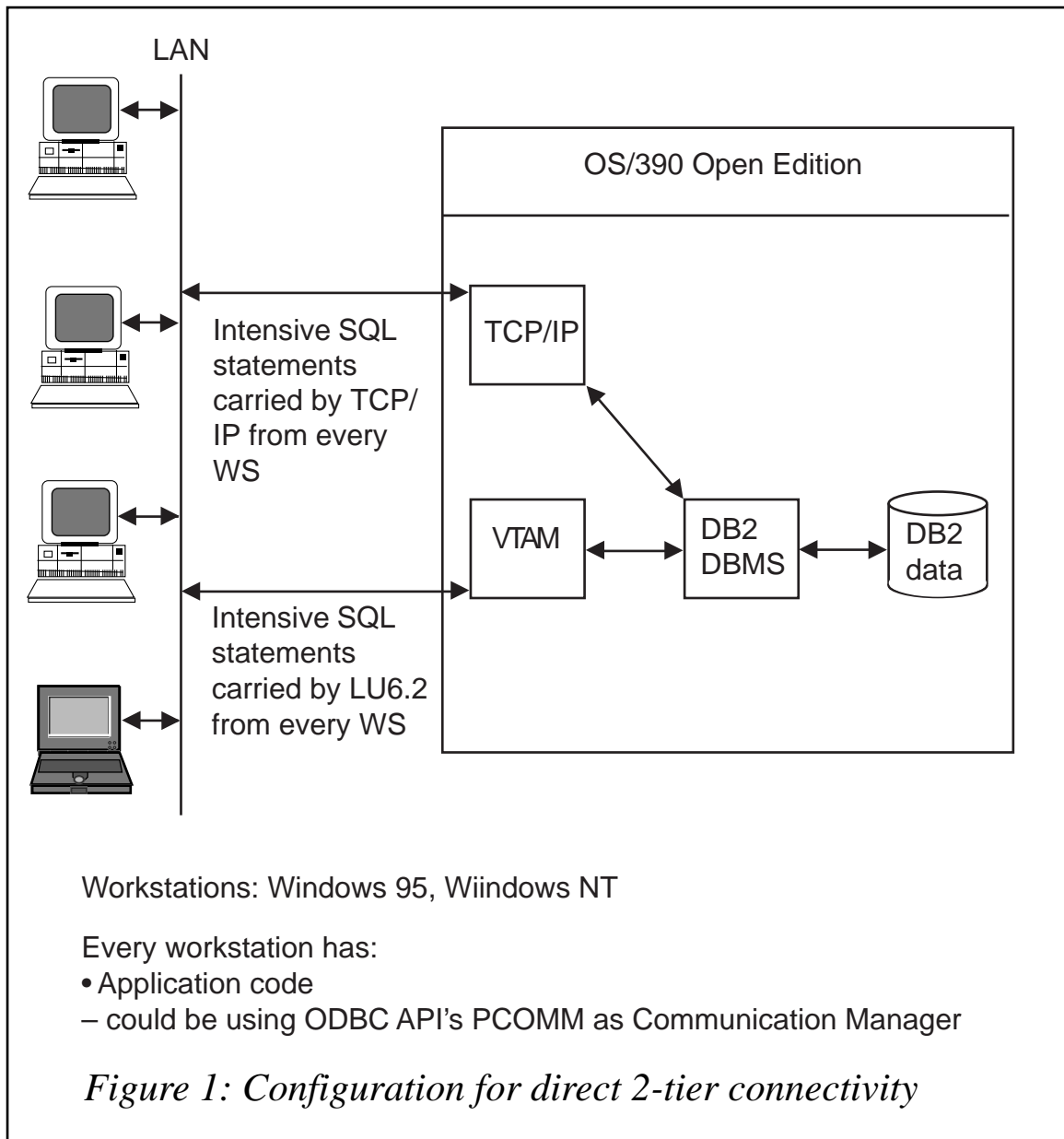
Direct 2-tier connectivity from a workstation (such as Windows 95 or Windows NT) to UDB DB2 OS/390 is illustrated in Figure 1.

This option consists of several workstations joined together through a LAN that connects to OS/390 over T lines such as T1(1.544MBps).

This option requires DB2 Connect Personal Edition (PE) software on the workstation. It also requires Communication Manager software such as PCOMM to reside on the workstation. This option also requires that application code resides on the workstation or at least that the code is loaded at execution time on the workstation from a file server located on the LAN.

Because of the location of the executables of application code on each client, the client is sometimes called a 'fat' client.

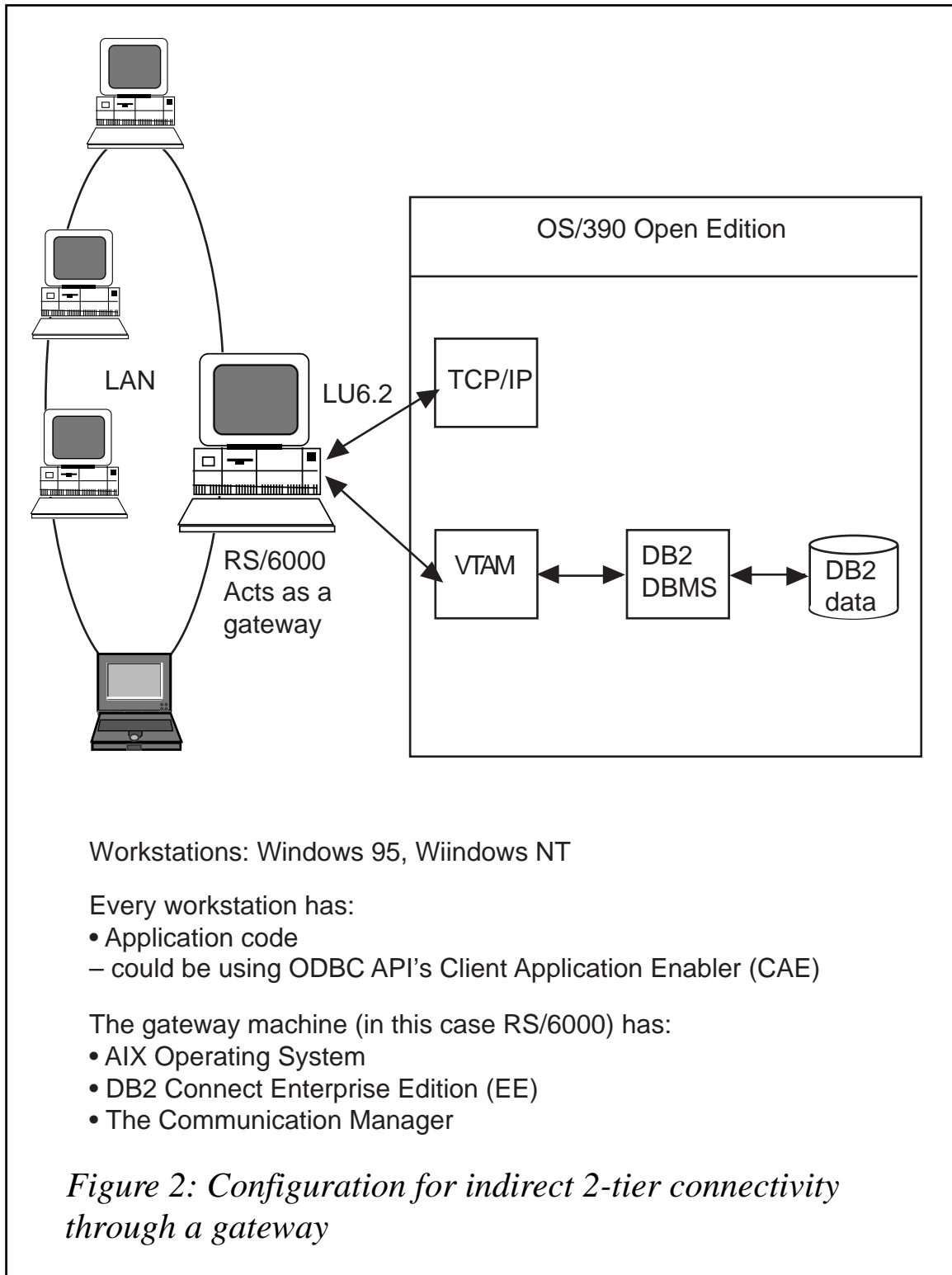
Application examples would be PeopleSoft ERP, SAP ERP, Power Builder, Visual BASIC, Microsoft Access, and Excel applications, etc. These applications use ODBC APIs to communicate with the host.



The ODBC calls are passed through the DB2 Connect (PE) which in turn, depending upon the protocol used in communication (TCP/IP or LU6.2), may translate the ODBC calls and pass them to the Communication Manager software, which in turn packages these calls and sends them through the communications lines to OS/390. Once the transmitted package is received, either by VTAM (if the protocol used is SNA) or by TCP/IP (if TCP/IP protocol is used), then it is passed to DB2 databases.

Indirect 2-tier connectivity from a workstation to UDB DB2 OS/390 via a gateway such as RS/6000

Indirect 2-tier connectivity from a workstation (such as Windows 95



or Windows NT) to UDB DB2 OS/390 via a gateway such as RS/6000 is illustrated in Figure 2.

This option consists of several workstations joined together through a LAN that connects to a gateway on the same LAN. In our case the gateway is an RS/6000 machine with AIX as the operating system. This gateway contains DB2 Connect Enterprise Edition (EE) and the AIX Communication Manager. (Note that if the protocol used from the gateway to OS/390 is TCP/IP then there is no need for the AIX Communication Manager.)

The application code still resides on every workstation. The Client Application Enabler (CAE) also resides on every workstation. The ODBC calls are passed by the Router in CAE to the gateway RS/6000 through the LAN protocol, which happens in this case to be TCP/IP.

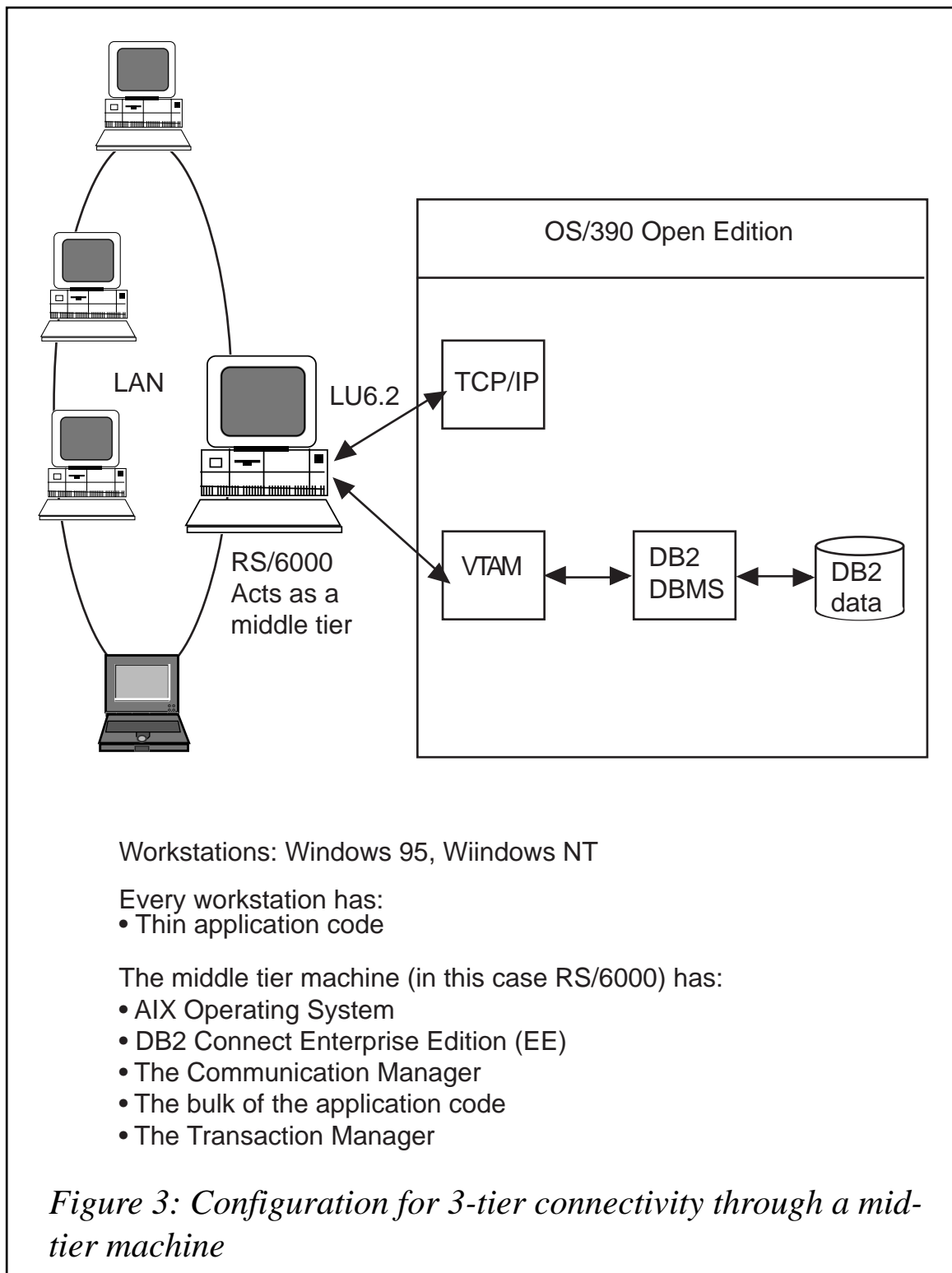
The flow of communication then goes through DB2 Connect EE and Communication Manager in the gateway to OS/390 over T lines such as T1.

3-tier connectivity from a workstation to UDB DB2 via an application server such as RS/6000

3-tier connectivity from a workstation (such as Windows 95 or Windows NT) to UDB DB2 OS/390 via an application server such as RS/6000 is shown in Figure 3.

This option consists of several workstations joined together through a LAN that connects to an application server machine. In our case the application server is RS/6000 (it could be NT) with AIX as the operating system. This application server machine contains DB2 Connect Enterprise Edition (EE) and the AIX Communication Manager. (Note that if the protocol used from the application server to OS/390 is TCP/IP then there is no need for the AIX Communication Manager.)

The application server also contains the bulk of the application code. To flash back: remember that in 2-tier connectivity the application code resides on the client and the client was called 'fat' client. Now the bulk of application code is stripped from the client (which is now



called a 'thin' client) and placed in the application server machine. Note that the Presentation Manager (the 'GUI' stuff) is still on each client machine.

The application 'stub' on each client packages application services through ODBC calls and sends it via the LAN through TCP/IP to the application server machine.

On the application server machine there is yet another piece of Software called Transaction Manager. The job of the Transaction Manager is to receive, accept, log, and manage the business requests sent by clients (see *IBM DB2 Connect EE Quick Beginnings V5* on the responsibilities of the Transaction Manager). An example of a commercial Transaction Manager would be Tuxedo from BEA.

The Transaction Manager then passes the requests to DB2 Connect EE and to Communication Manager (if SNA protocol is used) and then the packaged requests go through T1 communication lines to OS/390.

For best performance of DRDA these communication lines could be ESCON connections. We shall talk about that another time.

4-tier connectivity from a workstation to UDB DB2 OS/390 via the Web through an application server such as RS/6000

This connectivity option essentially is three-tier connectivity where clients can connect from the workstation to DB2 OS/390 via a Java-enabled Web browser that resides on the workstation.

The Web configuration will not be discussed in this article except to say that this connectivity needs a Web server, which could reside outside or inside the application server of the 3-tier set-up.

DB2 CONNECT

In all our configurations stated previously in this article, the DB2 Connect product was an integral part of the configuration. Please revisit Figures 1, 2, and 3 in this article to see where DB2 Connect is situated in the configuration.

So what is DB2 Connect? What is its purpose? And what does it do?

DB2 Connect is a software product from IBM that is designed to

facilitate connections between client applications and other related DBMSs that happen in our case to be DB2 on OS/390.

DB2 Connect used to be called Distributed Database Services (DDCS) in single and multi-user versions.

DB2 Connect comes in two flavours – DB2 Connect Personal Edition (PE) which connects only one user to OS/390, and DB2 Connect Enterprise Edition (EE) which connects multiple users to OS/390.

To facilitate connection from the client application to DB2 OS/390, DB2 Connect performs all sorts of functions. Here are some of them:

- Converts ODBC calls (if used by the application) to DRDA calls.
- Performs EBCDIC/ASCII character conversion.
- Creates an SNA or TCP/IP conversation with DDF.
- Sends requests and receives replies from client computers and the DDF.

For TCP/IP connectivity, DB2 Connect requires the following conditions:

- DB2 UDB for OS/390 Version 5.1 or higher.
- TCP/IP 3.2 or higher.
- OS/390 Version 1.3 or higher.

DB2 Connect needs three directories:

- The node directory.
- The system database directory.
- The Database Connectivity Services directory, also known as the DCS directory.

These three directories could reside in a skeleton instance if a local database is not needed, or could reside in a full instance database if the application logic needs local database processing either on the client or on the application server.

Let us examine some of the important entries in each directory and see how these directories can be updated with the necessary information.

Some of the important entries in the System Database Directory are:

- Application alias name.
- Communication Node name:
 - For SNA it would be the communication node name of OS/390.
 - For TCP/IP it would be the local node name of DB2 Connect WS.
- Authentication parameters such as DCS.
- Type of database entry – whether it is remote or local

One can see what is already defined in the System Database Directory by issuing the following command from the command line interface of the instance:

```
Db2> list database directory > db_directory.txt
```

or outside the command line interface by issuing the following command in AIX:

```
$db2 list database directory > db_directory.txt
```

The Database Directory can be updated in either one of the following two ways.

By the GUI Client Configuration Assistant as follows:

```
START, PROGRAMS, DB2 for Windows,CCA, push Add button etc.
```

In AIX by issuing the following command:

```
$db2 catalog database <local_db_name> as <name of the database alias> at  
node <node_name of OS/390> authentication DCS  
/* for SNA connectivity */  
$db2 catalog database <local_db_name> as name of the database  
alias>at node <node_name of DB2 Connect ws> authentication DCS  
/* for TCP/IP connectivity */
```

If one makes a mistake in cataloguing the alias, one can uncatalogue

the entry by using the following command:

```
$db2 uncatalog database <name of the database alias>
```

Note that in the Database Directory one can catalogue several aliases for the same physical target database, so several entries can be in this directory referencing the same target database.

Some of the important entries in the node directory are:

- Communication Node name:
 - For SNA it would be a communication node name of OS/390.
 - For TCP/IP it would be the local node name of the DB2 Connect WS.
- If SNA is used:
 - The side information name of the communication server.
- If TCP/IP is used:
 - The protocol used is TCP/IP.
 - The servicename.

One can see what is already defined in the node directory by issuing the following command from the command line interface of the instance:

```
Db2> list node directory > node_directory.txt
```

or outside the command line interface by issuing the following command in AIX:

```
$db2 list node directory > node_directory.txt
```

The node directory can be updated by either one of the following two ways.

By the GUI Client Configuration Assistant as follows:

```
START, PROGRAMS,DB2 for Windows,CCA,push Add button etc.
```

In AIX by issuing the following command:

```
$db2 catalog appc node <name of the OS/390 node> remote <the
                                sym_dest_name> security program
                                /* for SNA conn. */
$db2 catalog TCPIP node <name of the DB2 Connect ws> remote <OS/390 host
name or address> server <port_num_or_service_name>
                                /* for TCP/IP conn */
```

If one makes a mistake in cataloguing the node, one can uncatalogue the entry by using the following command:

```
$db2 uncatalog node <node_name>
```

Note that in the node directory one can catalogue only one node per machine: ie there will be only one entry in this directory per machine unlike the several aliases for the same physical target database in the system database directory that we mentioned before.

Some of the important entries in the DCS directory are:

- Application alias name.
- The name of the target DB2 OS/390 as it is stated in the communication record in the BSDS of the host DB2.

One can see what is already defined in the DCS directory by issuing the following command from the command line interface of the instance:

```
Db2> list dcs directory > dcs_directory.txt
```

or outside the command line interface by issuing the following command in AIX:

```
$db2 list dcs directory > dcs_directory.txt
```

The DCS directory can be updated by either one of the following two ways.

By the GUI Client Configuration Assistant as follows:

```
START, PROGRAMS,DB2 for Windows,CCA,push Add button etc.
```

In AIX by issuing the following command:

```
$db2 catalog dcs database <name of the local database> as <OS/390
```

```
target database as stated in the BSDS>  
/* this applies to SNA and TCP/IP protocol */
```

If one makes a mistake in cataloguing the DCS entry, one can uncatalogue the entry by using the following command:

```
$db2 uncatalog dcs database <name of the local database>
```

Note that in the DCS directory one can catalogue several aliases for the same physical target database, so several entries can be in this directory referencing the same target database.

We discussed DB2 Connect in detail. What we need to discuss now is how to install the DB2 Connect product.

Installation of DB2 Connect is very simple. The software comes on a CD-ROM. Mount the CD-ROM, and fill up the few fields on the few panels where prompted. Behind the scenes the software will create a full or skeleton instance to house the three directories we talked about previously. During the installation process two major icons will be created for you, namely:

- DB2 Client Configuration Assistant.
- DB2 Control Center.

Use the DB2 Client Configuration Assistant to populate the three DB2 Connect directories we talked about previously. Simply fill up a few fields within a few tab menus. Note that if the platform is NT, the installer ID should have administrator privileges.

After the three directories of DB2 Connect are updated with relevant information as was explained above, various database utilities such as the Command Line Processor and the DB2 CLI etc need to be BOUND to the database to be accessed. All these utilities have associated BIND files, but are usually lumped together per platform in a .LST file. In our case the platform is MVS. One could BIND these utilities using a push-button labelled BIND in the Client Configuration Assistant (CCA) or if you are using AIX one needs to issue the following BIND command:

```
db2 bind
```

```
INSTHOME/sql1lib/bnd/@ddcsmvs.lst blocking all sqlerror continue messages  
bind.msg
```

The INSTHOME (instance home) can be found by interrogating smit in AIX. It could be something like this: /sys2/home/db2devp1/.

One last remark about DB2 Connect: Client Application Enabler (CAE) was mentioned in this article when we discussed the indirect two-tier connectivity. The Client Application Enabler is really a subset of DB2 Connect. It needs two directories of its own. These directories are similar to the directories used by DB2 Connect. The two directories of CAE are system database directory and node directory. There is no DCS Directory for CAE.

Editor's note: this article will be concluded in next month's issue.

Nicola S Nur
Senior DBA (Canada)

© Xephon 2000

Call for papers

Why not share your expertise and earn soem financial reward at the same time? *DB2 Update* is looking to swell the number of contributors who send in technical articles, hints and tips, and utility programs, etc. We would also be interested in articles about performance and tuning, and information and tips for DB2 DBAs. If you have an idea for an article, or you would like a copy of our *Notes for Contributors*, contact the editor, Trevor Eddolls, at any of the addresses shown on page 2.

DB2 news

Sybase has announced its XJS/390 Enterprise Integrator and Mainframe Event Adapter mainframe application integration products.

XJS/390 is designed to speed creation of Web interfaces to mainframe processing and data. Developers can implement mainframe-based components in Mscript, a JavaScript-compatible language, or in XML.

It's for building new applications in an environment running on NT or Solaris. Developers create applications in Mscript that read and write to mainframe data sources and transactions. The tool includes class libraries and data type conversion tools to allow developers to work in Mscript even when interacting with COBOL transactions and mainframe data stores.

XJS/390 provides classes to interface XML and Mscript to DB2, VSAM, and MQSeries data queues.

Mainframe Event Adapter, meanwhile, provides integration of existing mainframe and other enterprise applications without changing the existing mainframe application code.

It's said to extend existing DB2 applications by detecting changes to DB2 data. It tracks updates, inserts, and deletes in DB2/MVS, generates events from defined changes to DB2 data, and sends event notification to other systems.

Available for OS/390 with support for DB2 and MQSeries OS/390, event definition

tools are available for NT and the Solaris Java Runtime Environment.

The XJS/390 applications are available either as options to Sybase Enterprise Portal or as stand-alone products.

For further information contact:
Sybase, 6475 Christie Ave, Emeryville, CA 94608, USA.
Tel: (510) 922 3500.
URL: <http://www.sybase.com/products/eaimiddleware/xjs390enterpriseintegrator>.

* * *

IBM has announced a range of new and tweaked database tools for its DB2 Universal Database for OS/390. Among these are database administration tools to help maximize availability, performance management tools, recovery and replication management tools, and application management tools.

Among the administration tools for DB2 are the Administration Tool itself, plus Forms, High Performance Unload, Automation Tool, and Log Analysis Tool.

New and repositioned performance tools are Performance Monitor, SQL Performance Analyzer, and Query Monitor. Recovery and replication management is addressed with DB2 Recovery Manager, Row Archive Manager, and DataPropagator.

For further information contact your local IBM representative.



xephon