IBM DATABASE 2 AIX/6000 FAMILY OF PRODUCTS

March 18, 1993

Meri Ortali
Leon Katsnelson
Connie Roakes DaFoe

Database Technology Product Planning
IBM Canada Lab
T/L 778-2378

IBM

The following technical description is an excerpt from the New Sales Manual Text section of RFA No. 20792, dated March 16, 1993. It includes an overall technical description of the DB2/6000 AIX family of products.

This information is being provided as a foundation document for learning about DB2/6000, its architecture, and heritage, as well as the new functions that will be included in this release.

Contents

Today′s database management systems environments are changing. The demand for information access is increasing as the growth of client/server systems continues in both large and small businesses. Users who were once satisfied with simple file manager programs now need powerful database management tools to manage and control their data. Application developers are requiring the features of database management systems once available only on larger machines. The complexity of the data and the variety of applications on top of this data is growing exponentially.

Customers are making decisions regarding downsizing and rightsizing of their systems and determining the best way to provide access to corporate data. These decisions include deciding what database applications should be kept on the host, what database applications should be duplicated or moved to the LAN, and where new database applications should be written.

More and more industrial strength, mission critical, line of business applications are moving to client/server LANs because of the advances in workstation hardware capacity and in software function and performance. At the same time, there is a large amount of legacy data that must continue to reside on host systems and be accessible.

IBM DATABASE 2 AIX/6000 Version 1 (DB2/6000) and IBM AIX Distributed Database Connection Services/6000 Version 1 (DDCS/6000) provide solutions for creating, storing, accessing, and updating data to meet the criteria and environments described above. Relational databases can be stored on a user′s personal workstation or at a database server that is part of a LAN. Additionally, connectivity to IBM′s host relational databases is provided. To the user, the host database can be viewed as just another server. DB2/6000 and DDCS/6000 are natural extensions of the AIX platform which currently also includes application development languages and tools, LAN services, system management, and communications products -- all available from one vendor -- IBM.

The following paragraphs provide a technical overview of the DB2/6000 database management system, followed by an indepth description of the key functions available in this version.

Industrial Strength Relational Database Management System

IBM DATABASE 2 AIX/6000 Version 1 is a relational database management system. It includes Database Services, Remote Data Services, application programming interfaces, database management and administration tools, a database Command Line Processor and a database client Call Level Interface. These components and their functions make IBM DATABASE 2 AIX/6000 Version 1 an industrial strength database management system for use in personal productivity, performance-oriented, mission critical, on-line transaction processing, and line-of-business applications.

IBM DATABASE 2 AIX/6000 Version 1 has a strong affinity with IBM′s host relational databases. It makes extensive use of the database technology found in IBM′s Database 2 for MVS (DB2 for MVS) and Structured Query Language/Data Systems (SQL/DS) host relational database products. In particular, IBM DATABASE 2 AIX/6000 Version 1 is designed to ensure database consistency, integrity, recovery, manageability, performance, and scalability in a standalone and client/server environment.

Technical Summary

With the announcement of DB2/6000 IBM now provides a family of Relational Database Management Systems available MVS, VM, VSE, OS/400, OS/2 and AIX RISC System/6000 platfoms. IBM′s commitment to providing high quality database products on established, stable platforms means customer investments will be justified again and again, as IBM enhances the relational database family members over time.

DB2/6000 provides a strong affinity with IBM′s host relational database management systems by providing a highly compatible SQL language interface. DB2/6000 makes extensive use of the database technology components found in DB2 for MVS and SQL/DS host relational database products. For example:

- These products all evolved from System R.
- They use a cost-based optimizer to determine the most efficient access route to data.
- They have an integrated catalog, that cannot be bypassed, to ensure data security and integrity.

IBM DATABASE 2 AIX/6000 Version 1 is based on and provides many of the same functional capabilities available in IBM DATABASE 2 OS/2 (DB2/2). System programmers will be able to apply the same set of database skills to both the OS/2 and AIX RISC System/6000 platform. Application portability, interoperability and asset protection are important benefits. In addition to providing many of the features available in DB2/2, DB2/6000 also provides support for a number of other key functions.

- Additional DB2 for MVS Compatibility
  - Value Scalar Function
  - Concatenation
- On-line backup and support for IBM Data Facility Distributed Storage Manager (DFDSM(TM)).
- AIX client/server support including
  - Unequal code page support
  - TCP/IP protocol support
  - APPC protocol support
  - DOS/WINDOWS client enhancements
- Call Level Interface
- XA support through CICS/6000
- Force Individual users
- Import Commit Count
- Database System Monitor API
- Graphical DBA Utilities
- Compound SQL
- On-line documentation (InfoExplorer format)

Database Engine Description

Database Services provides all of the base functions of the IBM DATABASE 2 AIX/6000 Version 1 and is commonly referred to as the database engine. All access to a database is performed by Database Services, and it is used to create databases, tables, views, indexes, recovery log files, and configuration files.

Database Services manages the data stored in the database, generates packages (access plans to access the data), and provides the database transaction management, data integrity, and data protection services. Database Services supports the Structured Query Language (SQL) which includes statements used to define, update, and control data in a database. Database Services also provides the utility functions that help the user maintain the contents of the database. Database Services transforms an SQL statement into a sequence of internal operations that optimize data retrieval. DB2/6000 uses a cost-based optimizer to determine the most efficient access route to the data. SQL language is the relational database interface component of IBM′s Systems Application Architecture.

To write advanced, high performance applications for DB2/6000 requires knowledge of the SQL language. For users who are familiar with writing DB2 for MVS or SQL/DS applications, many of the SQL statements are familiar or the same. The goal of each successive release of DB2/6000 is to make its SQL language statements, where reasonable and applicable, be more consistent with those statements used in our host databases, with the primary focus being compatibility with DB2 for MVS.

Both static and dynamic SQL statements are supported. Static SQL statements are created when the application program is written and the statements to be used are known and always the same. Running the program does not change the statement. Dynamic SQL statements are created during the execution of an interactive application program. Using dynamic SQL, the SQL statement is not fully known when the application program is written. Some or all of the statement is generated during the program processing. The operations and definitions vary depending on user input.

A DB2/6000 SQL Call Level Interface (CLI) is provided for the database clients. Call Level Interface is an alternative to dynamic embedded SQL. These functions provide the application capability to connect to a target DBMS, issue Structured Query Language (SQL) statements, obtain results and status information, all through procedure calls. CLI can be viewed as a carrier of embedded dynamic SQL.

Aside from eliminating the need for precompilation, CLI is also positioned properly to enable a client server database architecture. The application programmer does not have to deal with the concepts of precompilation and can build applications to the CLI without knowing which target database will be used. An application programmer using the DB2/6000 Call Level Interface will be able to develop CLI applications for DOS, DOS Windows, OS/2 and AIX clients to access the DB2/6000 server, or alternatively, through the DDCS/6000, the IBM host databases (DB2 for MVS, SQL/DS and OS/400). This CLI implementation is designed such that the CLI code is decoupled from the actual database manager and acts as a driver to the various IBM relational database servers.

IBM DATABASE 2 AIX/6000 Version 1 supports many of the same data types as those supported by IBM host relational databases. These include:

- Integer
- Floating Point
- Packed Decimals
- Fixed and variable length character strings
- Date, time, and timestamp

Date, time, and timestamp arithmetic and scalar functions are supported.

Database tables including these data types are defined using the SQL language. A table is a logical data structure consisting of rows (records) and columns (fields). The maximum number of columns that can be defined for a table is 255. The maximum length of a table row is 4,005 bytes. Special storage techniques allow a field in a table row to reference a character/bit string with a maximum length of 32,700 bytes. This allows a database to contain text, program code segments, image, graphics, or other data requiring a long field.

DB2/6000 supports declarative referential integrity. Referential integrity constraints can be declared when defining a table. Referential integrity ensures the consistency of data values between related columns in different tables. A referential constraint identifies a relationship between the primary key of a parent table and a foreign key of a dependent table. Referential constraints apply to insert, update, and delete operations that are performed on table data. DB2/6000 ensures that the values in the foreign key match the values in a related primary key. If they do not, the data cannot be inserted, updated, or deleted. Referential integrity′s implementation is consistent with DB2 for MVS implementation, both syntactically and semantically.

The use of referential constraints on databases can reduce development effort and improve the consistency with which constraints are applied. With referential constraints on a database, a database programmer can eliminate the special checks that would otherwise have to be programmed into each application accessing the database. The result is that these types of constraints need to be clearly defined only once in the database, simplifying the application programmer′s overall design.

DB2/6000 controls the security of your data. SQL GRANT/REVOKE statements help control user access to a database and the tables within a database by preventing unauthorized access.

Access to a specific database and the objects in it (for example, tables, views, access plans) is controlled by SQL GRANT/REVOKE statements. A creator, or other specifically authorized user of a database object (such as a systems administrator or database administrator), may protect the object by only granting access rights to specific users and/or groups. Another user must be specifically authorized to access and update a database object. These rights can also be revoked as required. A creator also has the option to allow public access to all database objects. SQL GRANT/REVOKE statements can be defined through the SQL interface.

Database Services support several callable environmental routines and configuration utilities. Environmental routines are used to create, drop, and recover databases. Configuration routines are used to display or change the DB2/6000 configuration file and individual database configuration files.

Database Services also provides utilities for backing up and restoring databases, importing and exporting data, reorganizing tables, and generating statistics on tables and indexes. Data can be imported and exported to and from a database in the following file formats:

Import File Formats:

- Non delimited ASCII
- Delimited ASCII
- WSF (Worksheet Format) used by Lotus
- PC IXF

Export File Formats

- Delimited ASCII
- WSF
- PC IXF

The user can specify a commit count to the database import utility. Import will perform database commits after the specified number of record imports have done. In the event of a crash or a log full condition during import, the user will not have to repeat the entire import operation. The import can be restarted from the point where the last commit was done.

The Backup database utility provides services to the user for creating a copy of the current database. This copy can later be restored using the Restore Database utility. Support within this utility will encompass the following:

- Off-line and on-line full local database backup
- Support for disk and tape media devices
- Support for remote backup to MVS or VM systemsd by utilizing IBM Data Facility Distributed Storage Manager (DFDSM).

  DFDSM is an enterprize storage management product that is based on a client server architecture that can be used to centrally manage the backup and archive of multiple workstations on a remote host. It supports various platforms and communications methods. It provides administrative facilities to manage the backup and storage of backup copies.

  On-line backup facility allows maximizes system availability by not requiring exclusive access to the database. Appliations can continue to access that portion of the database not in the process of being backed up.

DB2/6000 supports multiple instances (copies) of the DB2/6000 running on the same AIX RISC System/6000 processor. This function allows customers to use a single machine to both run production applications and to prototype, develop and test new applications without affecting the production environment. Customers can also use this feature to provide for more stringent security enforcement. For example, data that requires high level of security can be managed by one instance of DB2/6000 while other databases are managed by a second instance. In addition, multiple instances feature can be used to implement ″class of service″ concept by providing different configurations for different instances of DB2/6000. For example, one instance can be configured to provide for faster response by giving it more resources (buffer pools, higher process priority) while the other instance can manage less frequently accessed data or to support applications that do not the same level of performace. Each instance of DB2/6000 has a unique instance name associated with it. To connect to a particular instance a database user needs to set an environment variable to the name of the instance.

Integrity and Data Protection

Data integrity is essential in a relational database management system. Data integrity refers to the accuracy and credibility of the values within database tables. Whenever data is shared, there is a need to manage and control operations that ensure the integrity of the data in the database. DB2/6000 is based on the technology employed in DB2/2 and relies on research work performed for DB2 for MVS and other relational database products to provide high levels of data integrity. IBM has consistantly enhanced and optimized its support in this area.

- Transaction Management:

  DB2/6000 provides full transaction support in much the same manner as DB2 for MVS and SQL/DS. Any reading or writing of a database is done within a transaction.

  A transaction is a series of operations that may result in multiple updates to a database. COMMIT/ROLLBACK functions help insure that a database will be properly updated by an application transaction. Upon successful completion of a transaction, a COMMIT statement will be issued, which allows the database updates to become permanent. If an application is interrupted in the midst of a transaction, Database Services performs a ROLLBACK which backs out all tentative uncommitted updates after an application falure.

  Database Services performs transaction management through the use of locks and a recovery log. Logs are used to provide written entries for changes in tables and indexes. The lock function is used to prevent another application from reading a data record if a transaction is pending against that record.

  X/Open(TM) has defined a standard interface between a Transaction Manager and a Resource Manager for use in Distributed Transaction Processing. A single transaction in a Distributed Transaction Processing environment may occur across several resource managers. This is referred to as a global transaction. Support for a two-phase commit protocol is required to ensure integrity in the commitment of multiple resources. A bidirectional, system level interface, called the XA interface, has been defined, by X/Open, so that the work done at multiple resource managers can be correlated with a globally unique identifier.

  The XA interface is supported in DB2/6000. As a resource manager of a database resource, it can participate in a global transaction that is being coordinated by a XA compliant transaction manager. DB2/6000 will support IBM AIX CICS/6000 as the XA compliant transaction manager in the AIX RISC System/6000 environment.

- Concurrency Control

  Concurrent, multi-user application access to a database is supported by row-level (record-level) locking. Row-level locking ensures that an application maintains control of a database row until the transaction is complete. This prevents another application from changing a row simultaneously which could result in loss of data.

  DB2/6000 allows concurrent access to a database with three isolation levels that ensure data integrity through locking. The three levels that can be used to control locking and preserve the consistency of the data are:

  - Repeatable Read
  - Cursor Stability
  - Uncommitted Read

  Repeatable Read

  Repeatable Read holds record locks until the next commit point. Repeatable read keeps a lock on all records accessed by an application program since the last commit point. If the application reads the same record again, the values will not have changed. The effect of the repeatable read isolation level is that one application program can prevent other application programs or users from changing tables. As a result, overall concurrency may decrease.

Cursor Stability

Cursor Stability provides greater concurrent data access than Repeatable Read. Cursor Stability holds a record lock only while the cursor is positioned on that record. When the cursor moves to another record, the lock is released. This allows another application to read or to change rows of table data which the application with cursor stability has previously read. If the data is changed, however, the lock must be held until the data is committed. Cursor Stability applies only to data that is read. All changed data remains locked until either COMMIT or ROLLBACK is processed.

Uncommitted Read

A database application using Uncommitted Read will read and return all rows of data even if they contain uncommitted changes made by other applications. Uncommitted Read applies only to FETCH and SELECT INTO operations. For other operations Uncommitted Read performs the same functions as Cursor Stability. Because this isolation level does not wait for concurrency locks, overall performance may increase.

The application program makes the decision as to which data isolation level to use when the application is precompiled or bound to the database.

When concurrency is less of a consideration, an entire table can be locked to avoid the overhead of row-level locking. The LOCK TABLE statement locks an entire table. The lock is not released until the transaction is committed or rolled back.

Process model

A process model used by DB2/6000 is designed to increase data integrity and security. The database engine will run in a process and address space separate from that of the user application. As a result, application programs will be isolated from internal DB2/6000 shared data structures and SQL data files. This will make it virtually impossible for user applications to accidentally compromise the integrity of their own or another user′s data.

- System and Media Recovery

  Following a system failure or media failure, a restart of the DB2/6000 will automatically back out all uncommitted database changes and complete processing of all committed changes. To aid the user in recovering from media (disk/diskette) failure, DB2/6000 provides utility programs to backup and restore data. The Backup utility copies the entire database to the backup media. The Restore utility is used to reload a database that was saved using the Backup utility.

  DB2/6000 also supports Roll Forward Recovery and Restore Recovery functions as methods to protect the data in a database in case of system or media failure. This is especially important for mission critical applications.

  Roll Forward Recovery allows for the rebuilding of a database to a state beyond the time that the last backup copy was made. Changes made since the last backup are stored in a log. Using this method, a database is restored using the last backup copy and then the database logs are used to reapply the changes that have been made to the database since the backup copy was created. Using the Restore method, backup copies of the database are made at scheduled times. The backup copy of the database is only as current as the time that the backup copy was made.

Database Status Information

DB2/6000 includes a Database System Monitor. This snapshot application programming interface (API) provides the user with a vehicle to gather statistical information regarding the operation of the database. The snapshot monitor can be used in real time to assess the internal operation of the DB2/6000 The specific attributes monitored are:

- Database connections
- Locks
- Buffer pool activity
- SQL statement execution
- Sorting
- Cursors
- Number of deadlocks
- Table activity
- Unit-of-work status
- Database status
- Communications activity

This information can be used to:

- Adjust DB2/6000 configuration parameters
- Improve DB2/6000 performance
- Better understand DB2/6000 internal operations
- Debug applications
- Monitor usage of DB2/6000
- Provide statistical information

While some monitoring information will always be gathered by the Database System Monitor, most of the monitoring information can be selectively turned ON and OFF by the user. This gives users more flexibility to monitor only what is of most interest to them, without slowing the operation of DB2/6000 unnecessarily.

Performance

Performance is always a key objective in database management systems. Many technologies have been applied in this area, including optimization techniques, sort and join algorithms, and buffer-management techniques. Indexing and locking techniques also provide a high level of concurrency while maintaining high levels of performance.

Database Services includes a cost-based optimizer that evaluates an efficient way of retrieving data. Before the access plan of a query is generated, the optimizer first analyzes the query. Statistics kept in the system catalogs, such as the size of the table and the number of distinct values in an index, are consulted. Decisions on the actual access path, or strategy for accessing data and executing the query are then made. Important choices made by the optimizer include the following: whether indexes should be used, in what order a join should be performed, what join algorithm should be used, and whether and when a sort is appropriate. The optimizer contains the key technology enabling efficient query execution in the Database Manager.

Specialized sort algorithms are used for optimized performance in a relational database environment. The algorithm is very flexible in terms of RAM usage and temporary DASD storage is used only as necessary. The optimizer has a great deal of freedom in controlling sort operations based on overall performance and resource utilization considerations.

Stored Procedures implemented through the DB2/6000 Database Application Remote Interface (DARI) help boost performance by giving our customers the flexibility to execute routines that are stored at the database server. The Database Application Remote Interface allows a remote application program that uses a database to execute a routine that is stored at the location of the database. One procedure call to a remote database may represent several accesses to the database. Therefore, executing the preprogrammed procedure on the database server can reduce traffic over the communication network.

A Compound SQL statement provides the ability to group several related SQL statements into a single executable block. It is a subset of specifications defined in ISO/ANSI SQL3. The Compound SQL statement can be viewed simply as a way of blocking SQL statements (sub-statements). A single request is made to the application server. The sub-statements execute at the server with no response being sent back to the application until either they have all successfully completed or until an error is raised by any one of the sub-statements.

Client/Server and Host Database Connectivity Support

Client/Server architecture allows an application to split processing between client and server machines. Both DB2/6000 and DDCS/6000 products implement the client/server architecture by allowing a database client and a server to be run on the same RISC System/6000 machine or on an other machine on a network. In addition, client applications can execute on an AIX machine or a machine running a different operating system (DOS, DOS Windows, OS/2).

DB2/6000 and DDCS/6000 provide client support for the workstation on which they are installed. DB2/6000 client support allows local AIX applications to access either local databases or databases on a remote server. DDCS/6000 client allows local AIX applications to access host databases as well as databases on remote servers. In addition, remote clients are supported by installing DB2 CS/6000 product. DB2 CS/6000 allows remote clients running DOS, DOS Windows, OS/2 or AIX to submit requests to DB2/6000 or DDCS/6000. Remote clients can be workstations that have an appropriate DB2 Client Application Enabler installed (DB2 CAE/DOS, DB2 CAE/2, DB2 CAE/6000) or machines running DB2/6000 or DDCS/6000. Workstations running OS/2 ES 1.0 or DB2/2 also can act as remote clients using APPC communication protocol. DB2 CS/6000 base product supports clients connecting via TCP/IP protocol. In order to support clients connecting via APPC, the DB2/6000 workstation must have SNA Support feature installed.

Database clients can access databases on either local or remote servers. Client application programs do not need to know the physical location of the database or be concerned with the network protocol that is used for connecting to a particular server. This is accomplished by, first, cataloging (at the client) various servers workstations and then cataloging individual databases that the client will be accessing. Once a database is cataloged, it is assigned a symbolic name by which it will be known to all applications running on the client workstations. If either physical location or a communication protocol need to change at a later date, a user would simply recatalog either the server workstation or a database. No applciation changes will be required. If, for example, a client workstation has PRODUCTS database cataloged as residing on a DB2SERV1 node for development and test purposes but needs to work with a production database that is managed by DB2SERV2 node, a user can simply recatalog PRODUCTS database as residing on DB2SERV2 node once testing is completed. Similarly, if the user wants to use the APPC protocol instead of the TCP/IP to connect to the DB2SERV1 server node, he/she will simply recatalog DB2SERV1 as an APPC connected workstation. No changes to database application programms will be needed.

Client application enabling products, DB2 CAE/6000, DB2 CAE/2 and DB2 CAE/DOS, are available for use on AIX, OS/2, DOS, and DOS Windows workstations respectively, to permit access to DB2/6000 and other members of IBM Relational Database family of products. These client products, which are available separately, provide functions that are nessesary to enable applications to access a variety of IBM relational databases. They provide run-time support to applications that are developed using the following Database Services APIs:

- Full complement of SQL operations invoked via either embedded SQL or Call Level Interface
- Database Application Remote Interface Environment Command (DARI, also known as Stored Procedures)

- Import/Export API
- Programming interfaces for the binder
- Interrupt Environment API

In addition, DB2 CAE/DOS provides ODBC driver that provides run-time support for applications developed using Microsoft Open Database Connectivity Software Development Toolkit. This driver supports applications that conform to the Core API and Core SQL Grammar conformance levels.

DB2 Client Application Enablers also offer support that enables users to bind their applications to target databases, import and export data to and from several popular file formats, and to perform certain client administration functions. DB2 CAE/6000 also offers a Command Line Processor utility that allows for submitting ad hoc SQL queries, prototyping of SQL statements and use of certain database administration and environment commands.

DB2 Application Enabler products, when purchased, permit concurent use of the database client-enabling code on multiple DOS, DOS Windows, AIX or OS/2 clients within a single location (see location license). This allows of ease of license administration and flexible installation procedures. For example, Application Enablers can be installed on a numnber of workstations without ordering a separate license for each of the workstations, or, they can be installed on a file server on a LAN such as IBM LAN Server, Novell NetWare or on an NFS drive and used by any number of workstations at a particular location without purchasing any additional licenses.

DDCS/6000 product provides applications running on an AIX workstation with access to host RDBMS that implement DRDA Application Server (DRDA-AS) function. When installed on a RISC System/6000 workstation, DDCS/6000 provides this service to applications running on the same system. When used in conjunction with DB2 CS/6000 i.e. when running on the same machine, DDCS/6000 extends its services to client DOS, DOS Windows, OS/2 and other AIX workstations on the network. In this configuration, DDCS/6000 acts as a DRDA gateway and provides services to applications running on the same machine as well as applications running on various client workstations on a network. A request to access the host database is first passed to the DDCS/6000 gateway workstation from the DOS, Windows, AIX, or OS/2 clients. The DDCS/6000 workstation, in turn, manages the database protocols and routes the read/write requests to the host database.

Because DRDA specification requires use of LU 6.2 communication protocol, DDCS/6000 requires IBM System Network Architecture Services/6000 Version 1.2 to be installed.

DB2/6000 supports clients running applications using a different code page (codeset) than DB2/6000 code page. This unequal code page support function allows OS/2, DOS, DOS Windows and AIX client applications to run in a wide range of code pages, and still be able to access a DB2/6000 database. For example, OS/2 clients can be running on the IBM-defined code page 850 and accessing a DB2/6000 database encoded on the ISO 8859-1 Latin-1 codeset.

Application Programming Interfaces and Support

Application programs can access data by either using embedded SQL statements, or by calling Call Level Interface functions. DB2/6000 provides for data definition, retrieval, update, and control operations through SQL. SQL statements fall into two categories:

- Data Definition Language (DDL) statements used to define and control access to database objects such as tables and views.

- Data Manipulation Language (DML) statements used to manipulate data stored in tables.

In addition to the SQL language, DB2/6000 provides Environment APIs that can be used to control DB2/6000 access, manage transactions, and support application development.

Programming Options

DB2/6000 supports application programs running on local and remote clients written in a compiled language. To develop a compiled language program that uses embedded SQL an appropriate precompiler must be available to process the SQL statements. Programs must be precompiled because compilers, generally do not recognize the embedded SQL statements. The precompiler replaces embedded SQL statements with statements that are acceptable in the source code program to a target compiler. The precompiler also analyzes the SQL statements and generates packages corresponding to the statements. The packages are compiled low-level representations of the SQL statements and contain optimized instructions on how to access the desired data most efficiently. A program with embedded SQL statements must be precompiled against a database before it can be executed against that database.

The precompiler can optionally generate a bind file which contains all the SQL statements in the source program and all the related information needed to generate packages. This separate bind process is useful for the creation of packages when a third-party application program is installed or when multiple databases are accessed. In such cases, the source program can be run through only the binding process without having to be completely precompiled every time.

The DB2/6000 product includes C and FORTRAN precompilers that allow for development of applications for AIX clients (local or remote). The C precompiler generates code that can be compiled with the IBM XL C compiler. The FORTRAN precompiler generates code that can be compiled with the IBM XL FORTRAN compiler. When developing embedded SQL applications intended for use on DOS, DOS Windows and OS/2 clients, C and FORTRAN application code must be precompiled on an OS/2 workstation with the DB2/2 product installed and then compiled by an appropriate compiler on an OS/2 or DOS workstation. The C precompiler provided with DB2/2 generates code that can be compiled with IBM C Set/2 Version 1.0 to produce OS/2 applications. It also generates C source code that can be compiled with Microsoft C Version 6.0 compiler to produce application programs for DOS and DOS Windows clients. The FORTRAN precompiler generates code that can be compiled with WATCOM F77/386 Version 9.01 (Patch Level D) compiler to produce applications for execution on OS/2 clients.

Application programmers that wish to use COBOL programming language can use Micro Focus COBOL Version 3 compilers and tools and utilities supplied with these compilers to develop applications for AIX, OS/2, DOS and DOS Windows clients.

In addition to embedded SQL, application programmers can use Call Level Interface to produce application programs. Applcations developed using the SQL Call Level Interface do not require precompilation step. DB2/6000, DDCS/6000 and all client application enabling products (DB2 CAE/DOS, DB2 CAE/2, and DB2 CAE/6000) include libraries and header files required for development of CLI applications with the IBM XL C, IBM C Set/2 Version 1 and Microsoft C Version 7.0 compilers. Developers of applications for DOS Windows clients that wish to use ODBC driver provided by the DB2 CAE/DOS product should use Microsoft Open Database Connectivity Software Development Kit Version 1.0.

To provide client/server database solutions for our customers, early versions of DB2/6000 and DDCS/6000 have been given to a number of third-party vendors and IBM business partners. These vendors will develop programs which include application development tools, end-user applications, database administration tools, and vertical applications.

Using Other Languages and Compilers: DB2/6000 applications can be written in other languages not supported by DB2/6000 if additional steps are taken by the application developer. An appropriate precompiler must be written to process the SQL statements. The *IBM DATABASE 2 AIX/6000 Programming Reference* book includes instructions on how to write a customized precompiler using the Database Precompiler API.

Performance Guidelines in Application Creation

Applications accessing remote databases stored on a database server on a LAN can make use of three database features that can improve performance. These features are Record Blocking, the Database Application Remote Interface (DARI), more commonly referred to as stored procedures, and Compound SQL.

Record blocking can be used to limit network traffic by returning a block of rows to an application processing a remote database versus individual rows returned one at a time. Record blocking operates when a read-only cursor returns multiple rows from a remote database.

The DARI, or stored procedures, allows an application running on a client to call a stored procedure on an DB2/6000 database server. Processing can be split between the database client and the database server on a supported network. The server procedure executes and access the database locally and returns the information to the client applications. Using the stored procedures approach can reduce network traffic, improve performance for server intensive work, and access features that exist only at a database server.

A Compound SQL statement provides the ability to group several related SQL statements into a single executable block. It is a subset of specifications defined in ISO/ANSI SQL3. The Compound SQL statement can be viewed simply as a way of blocking SQL statements (sub-statements). A single call is made to the application server. The sub-statements execute at the server with no response being sent back to the application until either they have all successfully completed or until an error is raised by any one of the sub-statements.

The *IBM DATABASE 2 AIX/6000 Pogramming Reference* provides the information necessary for developing a database applications.

DB2/6000 includes a sample program called EXPLAIN that assists in database performance analysis. The EXPLAIN utility describes the access path the optimizer will use when a statically bound package is run. Sorts, use of indexes, predicate use, lock intent, scan direction, etc. are all described for each section of the package. A DBA or application developer can use the EXPLAIN utility to evaluate and understand the access path during application development and testing phases, as well as when tuning the database for improved performance. Changes to an SQL statement can be evaluated with EXPLAIN to make more efficient use of DB2/6000 resources.

Database Administration Utilities

A set of database administration utilities are provided with DB2/6000. DBA utilities can be optionally installed and provide functions for managing DB2/6000 itself and its databases. These functions include a:

- Database system and individual database configuration
- Data backup and recovery
- Node and database directory management

Configuration

The configuration functions provided allow the user to specify the resources allocated in DB2/6000 and in each of its databases. Examples include:

- Specifying the resources for the size of the buffer pool and log files that can be connected, and the size of the sort buffer.
- Specifying the number of concurrently active applications connected to a database
- Specifying more memory for storage management
- Enabling a database for roll-forward recovery

Recovery

The recovery functions allow the user to perform backup and restore operations including:

- Backing up the entire database
- Recovering (restoring and roll-forward recovery) a database, replace a database with a database of the same or a different name, or resume roll-forward recovery of a database after some portion of the recovery is done
- Restarting a database if auto-restart is not set.

Directory

The directory functions provide user with ability to define databases and specify their location. Functions are provided to:

- Create a new database
- Catalog a local or remote database
- Uncatalog a database whose access is no longer required
- Drop a database
- Catalog or uncatalog a workstation where a database resides
- List all the local and remote databases cataloged in the system database directory
- List all the workstations cataloged on a user′s workstation. These workstations are where the remote databases are located
- List databases that can be accessed when using the DDCS/6000 product to connect to host databases

Command Line Processor

The DB2/6000, DDCS/6000, and DB2 CAE/6000 products provide a Command Line Processor utility to allow users quick access to database functions from the AIX command line. The AIX command line processor is automatically installed when these products are installed. It provides access to both AIX and host databases. Experienced users familiar with SQL can execute SQL statements, database environmental commands, and database utilities directly from the AIX command line. These statements can also be issued in a UNIX shell script (for example Korn, Bourne, and C Shell). The command line processor can maintain a history file of all requests and redirect the input or output of the command line request. Help is also provided through the on-line help facility.

An example of some commands issued from the AIX command line follows:

```
DBM CONNECT TO PAYROLL
DBM SELECT * FROM USERID.EMPLOYEE WHERE DEPARTMENT = 32
```

In the environments where DDCS/6000 is available, the Command Line Processor can also be used to access host databases. SQL statements typed on the command line are routed by DDCS/6000 product to the host database where the statement is executed. The host database response is then routed through the DDCS/6000 to the requesting user. Because these requests are processed by the host database system, certain functions, for example backup/restore, are not supported when working with host databases.

National Language Support

The IBM DATABASE 2 AIX/6000 Version 1 provides National Language Support (NLS) by isolating the end-user interactions (menus and messages) into separate files that can be translated. The SQL language itself is not translated into national languages. However, provisions are supported for identifiers in SQL (table names, view names, and field names) to be defined in national languages. In addition, special consideration is given to functions that require special adaptation in different countries such as date/time formats, collating sequences, and monocasing rules.

DB2/6000 supports Double-Byte Character Set (DBCS) including the support of a mixed single and double-byte internal coded character set.

DB2/6000 supports clients running applications using a different code page (codeset) than DB2/6000 code page. This unequal code page support function allows OS/2, DOS, DOS Windows and AIX client applications to run in a wide range of code pages, and still be able to access a DB2/6000 database. For example, OS/2 clients can be running on the IBM-defined code page 850 and accessing a DB2/6000 database encoded on the ISO 8859-1 Latin-1 codeset.

Support for additional languages is provided in DB2/6000 and DDCS/6000. Specifically, DB2/6000 and DDCS/6000 will provide codepage support for the following:

- Latin-1 countries ISO 8859-1 registered in IBM as code page 819 for:
  - Australia, Belgium, Canada, Denmark, Finland, France, Germany, Iceland, Italy, Latin America countries, Netherlands, Norway, Portugal, Spain, Sweden, Switzerland, United Kingdom, and USA.
- Latin-2 countries ISO 8859-2 registered in IBM as code page 912 and with English MRI for the following countries:
  - Croatia, Czech, Hungary, Poland, Romania, Serbia (Latin), Slovakia and Slovania.
- Turkish ISO 8859-5 registered in IBM as code page 920 with English MRI
- Greek ISO 8859-7 registered in IBM as code page 813 with English MRI
- Israel ISO 8859-8 registered in IBM as code page 916 with English MRI
- Arabic countries IBM-defined code page 1046 with English MRI

English MRI is provided for:

- The following Latin-1 countries:

  Australia, Belgium, Canada, Denmark, Finland, France, Germany, Iceland, Italy, Latin America countries, Netherlands, Norway, Portugal, Spain, Sweden, Switzerland, United Kingdom and USA.
- The following Latin-2 countries:

  Croatia, Czech, Hungary Poland, Romania, Serbia (Latin), Slovakia and Slovania.
- Turkey, Israel, the Arabic countries and Greece.

Complementary Products

Available in the same timeframe as DB2/6000 will be the IBM Query/6000 product.

IBM Query/6000 is a Query and Presentation tool for use with IBM DB2/6000 on the IBM RISC System/6000 to retrieve data from IBM′s OS/400, DB2 for MVS and SQL/DS relational databases. It is designed for use by non-computer professionals, such as managers, sales people and administrators, who need access to corporate, departmental and personal data to do their jobs effectively. The user of Query/6000′s point-and-click interface can avoid the need to understand database structures or SQL to translate raw data into vital information.

Query

The Query function of Query/6000 guides the user in constructing a query to access data in a relational database. The simple point and click user interface means that no knowledge of SQL is required, although an SQL interface is provided for advanced users.

The user of Query can :

- Create databases
- List the databases available

- List the tables and views within in a database
- List the reports and queries available

The information needed to solve a business problem is often spread across several different data tables.

Business users are frequently unaware of how much information is available to them, or how it is organised. Query enables the user to select the subset of information needed to solve a business problem from the mass of available data.

Once a query has been constructed, it can be saved for future use as a query specification. This enables a query to be run regularly, e.g. for a monthly analysis of sales results. Queries can also be built by one user for subsequent use by other users or groups of users.

Report

The Report function of Query/6000 provides a report writer which allows the user to create reports quickly and easily using data accessed directly in the database or via the Query function.

There are two basic types of report - detail and summary. The layout and design of the report are controlled by the user through the graphical user interface, so that a user can tailor a report to meet specific needs. Header and footer text can be included, totals, percentages, and averages can be calculated. When a report has been designed, it can be stored as a report specification for later use by the originator or other users.

Data Access

Using the Data Access functions of Query/6000, the user can read and write data held in IBM relational databases. Data can also be imported and exported in a range of personal computer formats.

Table Edit

The Table Edit function of Query/6000 helps the user to create, browse and edit relational data tables.

Please refer to RFA 20809 for more detail on IBM Query/6000.

Also available in the same timeframe as DB2/6000 are several vendor products. Further details will be available from the product vendors. The following representative products are detailed as follows.

BRIDGE/FASTLOAD provides the user with the ability to move data into a DB2/6000 database directly from an ASCII file. The file may have been created by another application or via a host database unload utility. As another example, the file may have have been created elsewhere then distributed to multiple database workstations or servers for ultimate loading into databases located on those workstations.

BRIDGE/FASTLOAD significantly reduces load time since it works directly at the file level and bypasses the DB2/6000 logging procedures. BRIDGE/FASTLOAD uses it′s own loading procedure to create the DB2/6000 ″.DAT″ and ″.LF″ files which comprise the data of a table. Since the logging procedure is bypassed, a backup is required immediately after to ensure integrity and recoverability after data has been loaded. It is also recommended that no users be connected to DB2/6000 when a load executes.

BRIDGE/FASTLOAD for DB2, used with IBM′s DB2 for MVS relational database management system allows the user to enter an SQL SELECT statement for execution on DB2 for MVS. The data extracted from DB2 for MVS is then formatted into DB2/6000 ″.DAT″ and ″.LF″ files on MVS. These files are then downloaded directly to a DB2/6000 server where they replaces the corresponding files within the DB2/6000 database. This procedure can significanty reduce load time since it creates the DB2/6000 files directly on the host. A backup is required immediately after a load and no users should be connected to DB2/6000 when the load process is occurring.

Micro Focus COBOL(TM) V3 will provide support for DB2/6000 in the AIX environment. This compiler supports the highest level of ANSI standard COBOL syntax, along with many other syntaxes, including those of IBM's mainframe COBOL compilers. Micro Focus COBOL provides access to many other functions and capabilities including interfaces to DB2/6000.

Micro Focus COBOL can be used to create powerful applications to run in client/server environment, as well as to downsize applications from mainframes or do development work for eventual return to a mainframe environment. The compiler is accompanied by several useful tools for Developers, including the Animator visual debugging tool, and is integrated with the IBM AIX SDE WorkBench/6000.

The Micro Focus COBOL Compiler, along with the supplementary tools can provide a powerful, flexible, and familiar environment for COBOL programmers.