# DB2 Universal Database
# Siebel 99

# White Paper

Version 1.1

Copyright IBM Corporation 2000.

This page intentionally left blank.

# Table of Contents

# Table of Contents

## Introduction

The purpose of this white paper is to provide a concise summary of the architecture of DB2 Universal Database and some details regarding the points of integration for the set of Siebel Enterprise Applications. The objective is to provide the reader with DB2 installation and configuration knowledge to implement a Siebel environment that uses DB2 as its relational database. This document should be used in conjunction with the Siebel Installation Guide and Release Notes for Siebel 99.

This document is based on Siebel 99.0.12 and DB2 Universal Database Version 5.2 for AIX. The recommendations provided in this white paper are based on feedback from IBM software developers who worked with Siebel Enterprise Applications and the Siebel 99 documentation. This document covers general DB2 administration issues. The reader should also be familiar with another relational database environment.

Note: All the information contained in this document is based on publicly available information as of January 15, 2000 and is subject to change. IBM® disclaims all warranties as to the accuracy, completeness, or adequacy of such information. IBM shall have no liability for errors, omissions or inadequacies in the information contained herein or for interpretations thereof.

## Getting Started with DB2 Universal Database Version 5.2

### Installing DB2 on AIX

The installation of DB2 Universal Database on AIX can be completed using various methods. The easiest and recommended installation method is to use the **db2setup** application. This script prompts for all necessary parameters to complete a successful installation. The version of AIX must be 4.1.4 or later. The command **oslevel** can be used to verify the AIX version.

The required installation path for DB2 on the AIX platform is **/usr/lpp/db2_0v_0r** (e.g. db2_05_02 for DB2 Version 5.2). Ensure that there is sufficient disk space within the file system before executing the **db2setup** installation script:

```
    cd /usr/lpp
    df .
```

The minimum disk space requirement for DB2 Universal Database Enterprise Edition Version 5.2 on AIX is 55 MB. Online documentation can take up to an additional 72 MB.

DB2 utilizes operating system facilities to administer users and groups and their associated passwords. Three sets of users and groups are created during execution of the **db2setup** application. The DB2 instance owner is a user who has full authority of all databases created within the instance. A DB2 instance is a logical server that is used to group related databases. The name of a DB2 instance on UNIX must match an existing UNIX user. A default instance called **db2inst1**.

DB2 stored procedures can be configured as *fenced*, which forces execution to occur in a private memory area outside DB2. If the stored procedures are configured to be *unfenced*, they may perform more efficiently as they execute within a DB2 process, but they may also cause DB2 server errors to occur. Stored procedures are configured to be fenced by default (recommendation is to keep them fenced for security and reliability purposes). A UNIX user should be created to act as the owner of the stored procedure memory area during execution. The default stored procedure user name created by **`db2setup`** is called **`db2fenc1`**. Siebel applications exploit DB2 stored procedures to update database statistics and to perform other tasks. Once DB2 has been installed successfully there should be three user IDs and groups. To start the DB2 instance (server) you simply logon using the DB2 instance user ID and issue **`db2start`**. The following message should be displayed : SQL1063N DB2START processing was successful. The installation program should have detected TCP/IP as an available communications protocol. To verify that it has been set properly, issue the command **`db2set`**.

A DB2 Administration Server instance will be created during installation. The default user **`db2as`** will be created and this user is the owner of the administration server instance. The administration server is a special instance used for managing local and remote DB2 servers. DB2 graphical tools including the Control Center and the Client Configuration Assistant utilize the DB2 Administration Server to satisfy requests. To verify that it has been started properly, issue the command **`db2admin start`**.

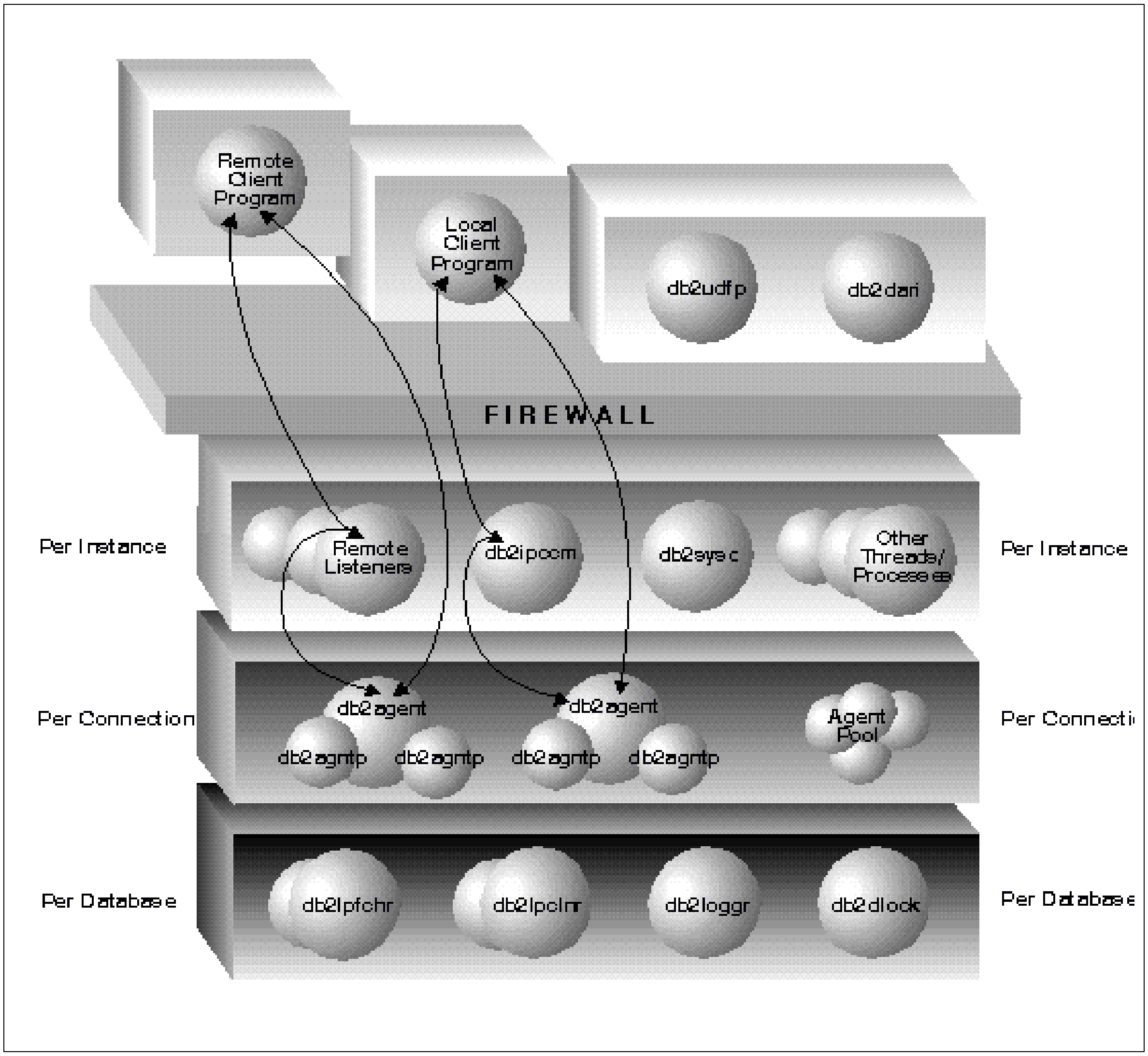
## DB2 Universal Database Architecture

A single UNIX machine can support multiple DB2 servers. Each DB2 server is known as a DB2 *instance*. A single instance can contain one or more databases. It is advantageous to have each database in its own instance from a configuration and tuning perspective because some of the resources are shared at the instance level. When most of the instance level configuration parameters are changed, the instance must be stopped and restarted. The instance (DB2 server) cannot be stopped (using the **`db2stop`** command) if there are any users connected to databases within the instance.

### Process Model

DB2 Universal Database on UNIX platforms is implemented using multiple processes. Each DB2 connection results in a new DB2 server agent process, called *db2agent*. Each DB2 instance has a single system controller process named *db2sysc*. To improve the performance of database connections, a number of preallocated or idle agents can be configured. If DB2 has been configured to utilize multiple CPUs for each client connection, known as intra-query parallelism, then the sub-agent process could work in cooperation with a single coordinator *db2agent*. In a Siebel environment most of the transactions can be handled efficiently by a single CPU, and the recommendation is to disable intra-query parallelism. The **`intra_parallel`** database manager configuration parameter is enabled by default if the hardware supports multiple CPUs, so this parameter should be changed if the DB2 server for a Siebel environment uses SMP (symmetric multiprocessing).

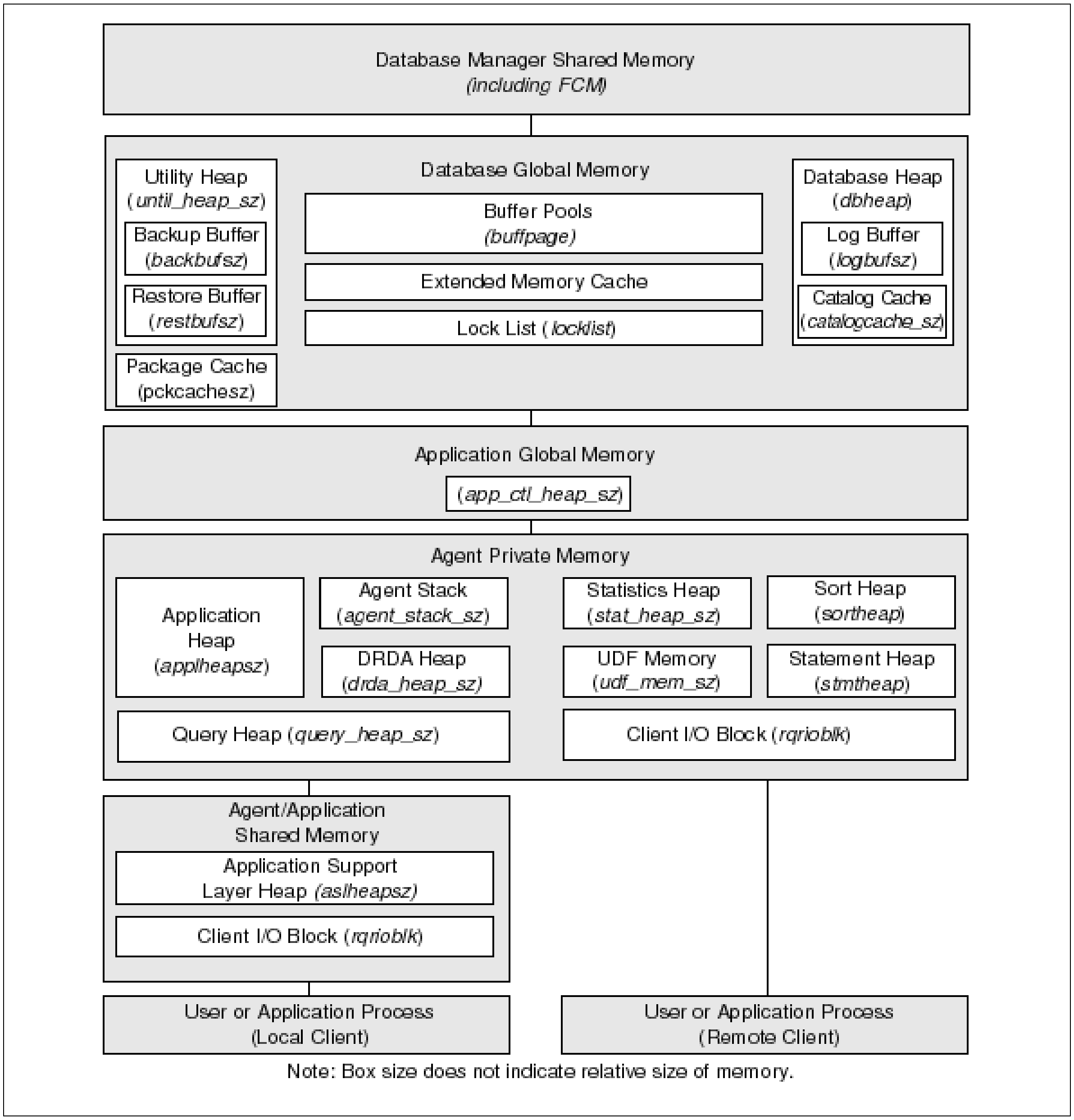
Once the first connection to DB2 is established, a number of database-supporting processes are created. These processes are responsible for transaction logging, deadlock detection, data prefetching, and other database tasks. The number of database-related processes is configured within the database configuration file. DB2 relies on the operating system process scheduler and does not directly influence which of its processes receives priority. Within a UNIX environment, such as AIX, the **`ps`** command can be used to determine how many processes have been allocated, and **`vmstat`** can determine memory and CPU usage.



## Memory Usage

By far the largest single parameter that affects memory allocation in DB2 is **`buffpage`**. This parameter represents the size of the data cache for each database. This memory area is allocated for each database and represents a multiple of 4KB pages (the default page size for DB2). Multiple buffer pools (data caches) can be allocated and are actually required for a Siebel/DB2 environment.

The following graphic shows the relationship of various DB2 memory parameters.



# Anatomy of a DB2 Database

## Database Sizing

Sizing a DB2 database requires knowledge of the number of tables and the expected number of rows. The average length of the row values are also required. Storage requirements vary depending on the data types involved.

The internal representations of DB2 data types are:

| | |
|---|---|
| DATE | 4 bytes |
| TIME | 3 bytes |
| TIMESTAMP | 10 bytes |
| DECIMAL (P,S) | depends on precision (p/2+1 bytes) |
| REAL | 4 bytes (32-bit representation) |
| FLOAT/DOUBLE | 8 bytes (64-bit representation) |
| SMALLINT | 2 bytes |
| INTEGER | 4 bytes |
| BIGINT | 8 bytes (not used by Siebel) |
| CHAR | 254 bytes |
| VARCHAR | 2000 bytes |
| LONG VARCHAR | 20 bytes internally (external to data page - 32,700 bytes) |
| CLOB | varies (external to data page - 2 GB) |
| BLOB | varies (external to data page - 2 GB) |

Siebel applications utilize various large objects (CLOBs). This data type does not get buffered through the DB2 buffer pool. It is recommended that a unique table space for large object data is created to improve performance. The large object table space should be created using preallocated files to take advantage of file system buffering capabilities.

## Physical Storage

A DB2 record can contain a maximum of 4005 bytes of regular data when a 4 KB page size is used. If an 8 KB page size is used, records can contain 8101 bytes of regular data. A maximum of 500 columns for each record can be defined for 4 KB pages, and 1012 columns for 8 KB pages. Regardless of page size, a maximum of 256 rows can be placed on a single page.

An 8 KB table space must be created because some of the application tables require more than 500 columns of data. The Siebel Enterprise applications has its tables grouped into appropriate DB2 table spaces.

DB2 Universal Database Version 6.1 has introduced 16 KB and 32 KB page sizes. At the time of initial publication of this document, Siebel Enterprise applications had not been tested using Version 6.1 of DB2.

Many of the physical storage parameters are set at the table space level and not at the table level. For example, the amount of free space to be left on each data page during table creation is a table space parameter. Before executing the **`install.bat`** script provided by the Siebel application environment, the table spaces should have been created using the names suggested in the Siebel 99.x Installation Guide.

The default insertion algorithm used by DB2 is to fill each physical page to 90 percent capacity. Clustering indexes can be defined within DB2 if the table access characteristics are well-known. A higher value for free space (**`PCTFREE`**) helps DB2 maintain the clustering of new data records onto existing pages. DB2 tries to place a newly inserted row onto the proper page, according to the clustering key, and if it will not fit, it tries to place the row nearby (in the same extent). If there is not enough free space, the new row is appended to the end of the table.

### Table Spaces

DB2 supports two types of table spaces: Database Managed Storage (DMS) and System Managed Storage (SMS). The default table space type is SMS, and it involves one or more file systems and its storage is extended as data is inserted or loaded into the table. DMS table spaces are recommended for most Siebel Enterprise application installations as the storage is preallocated. DMS table spaces may use *devices* (logical volumes) or *files* for its storage mechanism. The devices must be created by a UNIX administrator and sized properly before they can be defined for a DB2 database. All of the devices must be owned by the DB2 instance owner.

Table spaces consist of one or more storage areas called *containers*. Each container is divided into *extents* according to the value of the **`EXTENTSIZE`** table space parameter. Many tables could be allocated within a single table space. It is recommended that a relatively small number of table spaces be used for a Siebel application database; doing so provides for easier database administration and maintenance. Table spaces should be created using multiple container definitions to enable parallel I/O. Each container definition should correspond to a physically independent device.

### System Catalog

When a database is initially created, system catalog tables are created. These tables grow slowly as database objects and privileges are defined for the database. Initially, they use approximately 2.5 MB of disk space. The system catalog tables are grouped within the **`SYSCATSPACE`** table space. It is recommended to define this table space as SMS for ease of administration (the default DB2 table space is SMS).

### User Tablespace

Any user-defined tables are created within the default table space called **`USERSPACE1`**. Following initial database creation this table space will reside in the instance owner’s home directory. It is recommended to drop the default table space (**`USERSPACE1`**) and define the new table spaces as described in the Siebel Installation Guide utilizing multiple containers.

Four DMS table spaces are recommended. Character-based raw devices (logical volumes) should be used to define the containers for these table spaces. The table spaces must be created before the Siebel **`install.bat`** script is executed. Table space names are important because they are referenced in the **`install.bat`** during setup and installation of the Siebel Database Server.

### Temporary Tablespace

DB2 uses temporary table spaces for disk-based sorting activity and other tasks. By default, there is a single temporary table space called **`TEMPSPACE1`**, which is located in the instance owner's home directory. Usually this is not the location where sorting activity is desired, so two new temporary table spaces should be created on alternative storage devices. One of the temporary tablespaces should be defined for 4 KB objects and another for 8 KB objects. These table spaces should be created using System Managed Storage (SMS) and their usage can be monitored using standard operating system commands (e.g. **`du`** or **`dir`**). Once a second temporary table space has been created, **`TEMPSPACE1`** can be dropped.

### Buffer Pools

An 8 KB buffer pool must be created to support accessing the 8 KB table space data. The majority of the Siebel database tables are stored in 4 KB table spaces. The memory allocated for buffer pools is allocated during the first database connection and it is held until the last connection is released. The default buffer pool should be defined using the **`buffpage`** parameter in the database configuration. Buffer pool activity can be easily monitored using the DB2 snapshot monitor. Buffer pool definitions are stored in a system catalog view called **`SYSCAT.BUFFERPOOLS.`** To verify the existing table spaces, issue the following SQL statement: **`db2 SELECT * FROM syscat.bufferpools`**

### Recovery

DB2 will perform automatic recovery if the database server is interrupted. The database configuration parameter **`softmax`** may affect the length of time required to perform this recovery activity. This parameter determines the frequency of checkpoint activity to ensure that transactions have been reflected within the DB2 database. The value of **`softmax`** refers to the percentage of transaction log file space used prior to checkpoint activity.

### Transaction Logging

Transactions are managed by DB2 using a 'write-ahead' logging method. Activity is reflected within the buffer pool and in transaction log files. A newly created DB2 database will reuse the log files for transaction logging. This reuse mechanism is referred to as *circular logging*. If a database is damaged and a full recovery is required, the database must be configured to retain all log files. This technique is known as *archival logging*. Archival logging can be enabled by setting the database parameter **`logretain`** on. An additional parameter called **`user_exit`** can be enabled to automatically relocate non-active log files to an alternative location. Sample user exit programs can be found in the instance owner's home directory under the path `sqllib/samples/c`. The samples are called `db2uext2` with various file extensions.

### Database Backup Procedures

DB2 provides backup capabilities at the database level and the table space level. These backup images are individually time-stamped and they consist of database configuration, transaction log information, and all of the database objects. The backup can be created on tape devices or multiple disk files. Many DB2 installations use the ADSM (ADSTAR Distributed Storage Manager) product to coordinate database transaction log activity and database backup images.

### Database Restore Procedures

A DB2 database can either be restored in its entirety or have a subset of its table spaces restored. Any database activity that occurred since the backup image was created is applied using the transaction log files and the **`ROLLFORWARD`** command.

### Roll Forward Recovery

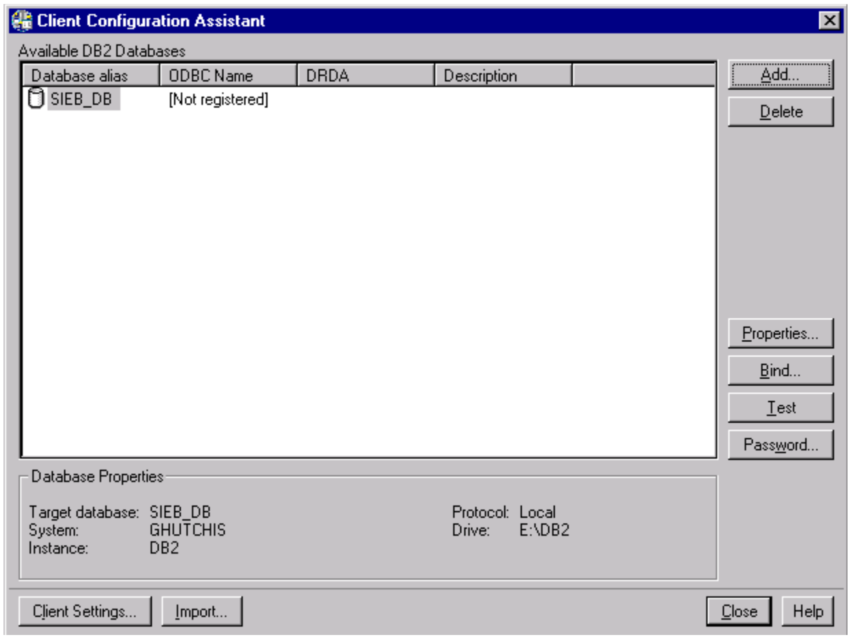
If a DB2 database has been enabled for archival logging (required for production environments) then any restored database image must have its transaction log files applied. Typically, all of the changes are applied using the following command.

```
db2 ROLLFORWARD DB sieb_db TO END OF LOGS AND STOP
```

## Siebel 99 Utilization of DB2 Universal Database

### Getting Started with Siebel 99.x

The Siebel connected client establishes a direct link to the DB2 Universal Database server. The Siebel Gateway Server does not interact directly with the DB2 Universal Database server. The Siebel Server communicates with the Gateway Server. A Siebel connected client must have ODBC access to the DB2 Siebel database. This can be accomplished using the DB2 Client Configuration Assistant from the workstation which has the Siebel Server installed. The **Add** push button should be used to catalog the DB2 database. The DB2 Administration Server on the server machine must be initialized for the **Add** operation to work successfully using the option **Search the Network**. The Client Configuration Assistant will initially connect to the DB2 Administration Server and a list of the DB2 instances currently available will be provided. When the proper database has been located the operation can be completed using the **Done** push button.



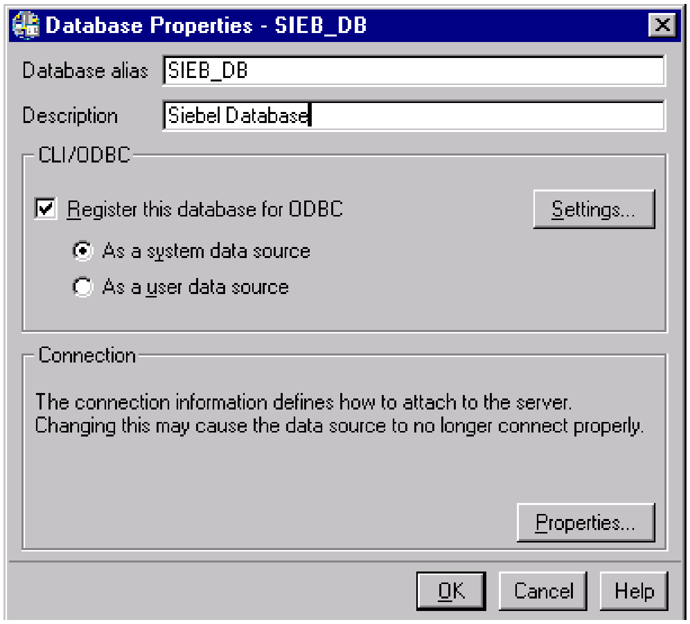
Available DB2 Databases

| Database alias | ODBC Name | DRDA | Description |
|---|---|---|---|
| SIEB_DB | [Not registered] | | |

Database Properties

Target database: SIEB_DB
System: GHUTCHIS
Instance: DB2
Protocol: Local
Drive: E:\DB2

The database must be registered as a system ODBC data source. The **Properties** button in the Client Configuration Assistant can be used to configure an existing database as an ODBC database source.



Database Properties - SIEB_DB

Database alias: SIEB_DB

Description: Siebel Database

CLI/ODBC

☑ Register this database for ODBC

● As a system data source

○ As a user data source

Connection

The connection information defines how to attach to the server. Changing this may cause the data source to no longer connect properly.

The network connection between the Siebel Client and the DB2 server should be verified using the following command: (must be issued in the DB2 Command Window)

```
    db2 CONNECT TO <db-name> USER <user ID> USING <password>
```

For testing purposes, use the DB2 instance owner name and its password.

### User Access for Siebel Applications

A group must be created on AIX called **`sse_role`**. Each Siebel user must have a unique user ID on AIX which is a member of the group **`sse_role`**. A user called **`siebel`** and **`sadmin`** must be created and they must belong to the **`sse_role`** group. Database privileges are provided to all members of this group. DB2 user names and group names must adhere to the following guidelines. Restricted user IDs include: USERS, ADMINS, GUESTS, PUBLIC, LOCAL, or any SQL reserved word listed in the DB2 SQL Reference manual. They cannot begin with: SQL, SYS, or IBM. The user names can include: A through Z, 0 through 9, @,#, or $. User names are currently restricted to a maximum length of 8 characters. A script called **`grantusr.sql`** is provided to grant Siebel users the proper database privileges.

### Completing the Installation

The Siebel environment must be set up for DB2 installations by editing and executing the **`siebenv.bat`** file appropriately. A file called **`install.bat`** provided with the Siebel Database Server must be modified to refer to the DB2 database. The SRC_USR and SRC_PSWD needs to be set to the Siebel Application user login and password. The SRC_TBLO value is the table owner. An ODBC data source name is required in this script. To simplify the installation, the DB2 ODBC data source name on the Siebel Server should be the same as the DB2 ODBC data source name on the Siebel Clients. Change the **`DATABASE_PLATFORM`** variable in the **`install.bat`** file to “DB2udb”.

The remainder of the **`install.bat`** script does not need to be modified. Definitions for all DB2 stored procedures and user-defined functions are created during the execution of the installation script. DB2 will also bind its required application packages.

Before executing the **`install.bat`** script, all of the required DB2 table spaces must exist and all of the recommended database and instance configuration changes must be complete. The DB2 installation section of the Siebel 99.x Installation Guide provides recommendations for table space creation.

## Application Interface Summary

### DB2 Call Level Interface (CLI)

The DB2 Call Level Interface (CLI) is a supported C language interface to DB2. The ODBC (Open Database Connectivity) interface is very similar to the set of DB2 CLI functions and therefore provides similar functionality.

### DB2 Embedded SQL

DB2 utilizes packages that contain static or dynamic SQL statements. These packages are stored in the DB2 database in the system catalog view (**`SYSCAT.PACKAGES`**) and they are updated using the **`BIND`** command when a new application library or module has been created. The **`install.bat`** script will bind any packages required by the Siebel applications.

### User-Defined Functions and Stored Procedures

When the Siebel Database Server is installed, a set of SQL extensions is created for the DB2 database. These SQL extensions are known as User-Defined Functions, or UDFs. UDFs are supplied as libraries and are unique to each operating system. They are located in the Siebel Database Server installation path `\db2udb\SiebProc\AIX`.

DB2 stored procedures are also supplied with the Siebel Database Server and must be installed (copied to the proper directories and defined) before the **`install.bat`** script is executed.

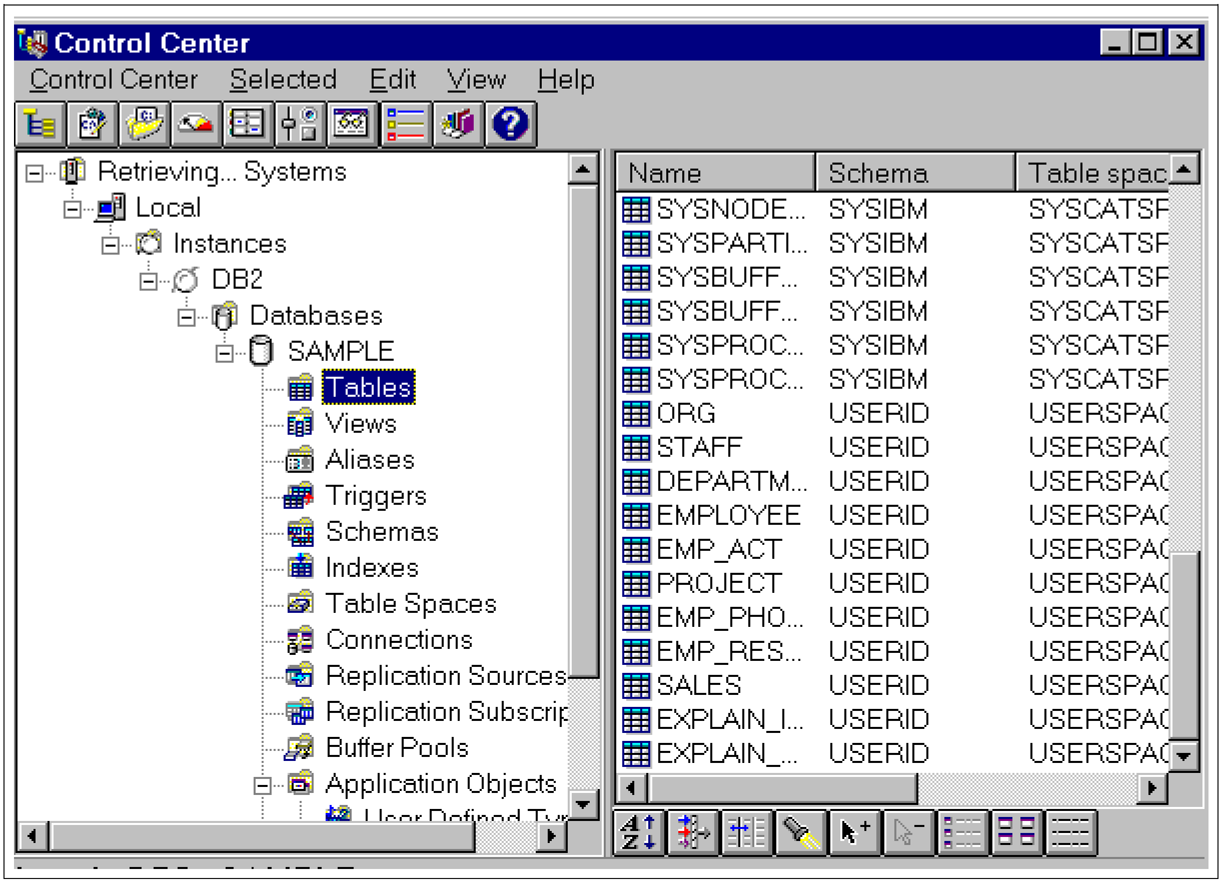
### Triggers

A trigger is a database object that will execute when modifications are made to the DB2 database. Modifications can include INSERT, UPDATE, or DELETE operations. There are 12 types of triggers available for use in a DB2 database. These triggers can executed for each row that is modified or once for each SQL statement. A trigger can also be defined to execute before the modification operation occurs. DB2 triggers are stored in the database an optimized DB2 when executed.

# DB2 Tools

## Control Center

DB2's Control Center is the primary graphical administration tool; it can be used to perform common database administration tasks such as creating database backup images, performing tuning, and query analysis.



| Name | Schema | Table spac |
|---|---|---|
| SYSNODE... | SYSIBM | SYSCATSP |
| SYSPARTI... | SYSIBM | SYSCATSP |
| SYSBUFF... | SYSIBM | SYSCATSP |
| SYSBUFF... | SYSIBM | SYSCATSP |
| SYSPROC... | SYSIBM | SYSCATSP |
| SYSPROC... | SYSIBM | SYSCATSP |
| ORG | USERID | USERSPAC |
| STAFF | USERID | USERSPAC |
| DEPARTM... | USERID | USERSPAC |
| EMPLOYEE | USERID | USERSPAC |
| EMP_ACT | USERID | USERSPAC |
| PROJECT | USERID | USERSPAC |
| EMP_PHO... | USERID | USERSPAC |
| EMP_RES... | USERID | USERSPAC |
| SALES | USERID | USERSPAC |
| EXPLAIN_I... | USERID | USERSPAC |
| EXPLAIN_... | USERID | USERSPAC |

## Creating a DB2 Database

Creating a DB2 database also creates a number of system tables (with indexes) and log files. By default a DB2 database will reside in the instance owner's home directory. You should specify a desired path during database creation and avoid creating the database in the instance owner's home directory. The command to create a DB2 database is shown below.

```
    db2 CREATE DB <siebel_db> ON <dbpath>
```

Three default table spaces are created: **SYSCATSPACE**, **TEMPSPACE**, **USERSPACE1**. The temporary table space should be moved to an alternative location to allow for better disk space management and monitoring. For example, a UNIX command such as **iostat** can be used to verify that the temporary table space is being used if the table space is located on a unique disk. The **USERSPACE1** table space is not used by Siebel applications and therefore should be removed.

A Siebel database requires four table spaces, called:
1. SIEBEL_4K
2. SIEBEL_8K
3. SIEBEL_4KL
4. SIEBEL_IDX

The SIEBEL_8K table space will be assigned to use the 8 KB buffer pool, which must also be created. The 8 KB buffer pool should be called SIEBEL_8K and should be sized according to the Siebel Installation Guide.

### User Management

The **SIEBEL** user (created in an earlier section of this document) must have the proper DB2 authority. To ensure that the proper authorities and privileges have been granted, ensure that the provided **grantusr.sql** script is executed:

```
    db2 CONNECT TO DB siebel USER <user_name> USING <password>
    db2 -tvf c:\sieb_install\grantusr.sql
```

### Siebel Repository

The **IMPREP.BAT** utility creates the Siebel repository. This script is only required during the initial setup of a Siebel database. The **DB_PLATFORM** variable must be set to "db2udb" and **SIEBEL** must be the owner of all tables.

### Data Movement

The **DATAEXP** utility provided with the Siebel applications should be used to extract data from the DB2 database. The **DATAIMP** utility is used to insert previously extracted data into a DB2 database. DB2 will log every transaction during the execution of these utilities. Therefore, it is recommended to ensure that commits occur frequently, using the options /x r (row) and /n y (rows per commit). The commit frequency should be determined by examining the available DB2 transaction log space (**`LOGPRIMARY/LOGFILSZ`**) parameters in the DB2 database configuration.

### Transaction Logging

The Siebel Installation Guide recommends 10 log files or 32 MB for transaction logging purposes. This is simply a guideline and should be adjusted if "log full" conditions occur. It is recommended to enable archival logging. This is accomplished by turning the **`logretain`** parameter on and configuring a user-exit program to relocate archival log files to an alternative location. The transaction log files should be moved from the default location to an new path. The log files can be relocated using the database parameter **`newlogpath`**.

### DB2 Query Performance

DB2 uses a cost-based optimizer that relies on statistical information to develop access plans for SQL statements. These access plans are stored as packages within the system tables if the SQL statement is associated with a static embedded SQL application module. Most of the SQL statements are analyzed dynamically and the plans are generated during application utilization. These access plans are cached in memory. The size of this cache is determined by the database manager configuration parameter **`pckcachesz`**.

The amount of optimization can be controlled using the database configuration parameter **`DFT_QUERYOPT.`** Supported optimization classes include: 0, 1, 2, 3, 5, 7, and 9. The default is set to 5, but for applications such as Siebel, the class should be set to a lower value of 1 or 2.

DB2 provides a graphical tool called Visual Explain, which can be used to help understand the chosen access plan for a given SQL statement. DB2's powerful query optimizer can rewrite SQL queries to improve the efficiency of its access plan. There are three major categories of rewrite operations that may occur including: operation merging, operation movement, and predicate translation. Most of the DB2 query rewrite rules are applied at optimization level 3.

Since DB2 is a cost-based optimizer, it is essential to update the statistical information for all of the active tables using the **`RUNSTATS`** command. The **`RUNSTATS`** command calculates various statistics on each table and the optimizer considers these statistics during access plan creation.

### Capturing Snapshot Monitor Data

DB2's snapshot monitoring facilities can be used to understand database utilization and tune DB2 to improve query performance. Snapshots can be obtained for databases, tables, table spaces, and locks. The best place to start is to take a snapshot of activity at the database level using the following command (substituting the name of the database for "siebel"):

```
    db2 GET SNAPSHOT FOR DATABASES ON siebel
```

## Analyzing Snapshot Data

Snapshot data can be used to help spot these events that can affect performance:

1. Locktimeouts
2. Lock escalations
3. Sort overflows
4. Package cache hits
5. Catalog cache hits
6. Asynchronous page reads (prefetchers)
7. Asynchronous page writes (page cleaners)

Application concurrency will be effected if lock escalations or timeouts occur. Lock escalations could occur if the database parameter **`locklist`** is set too small for the number of concurrent applications using the database. Sort activity occuring on disk can be monitored using **`iostat`** on the temporary table space location. If sort activity is overflowing to disk then a disk-based sort can be avoided by increasing the database parameter **`SORTHEAP`**. For more information on performance tuning refer to the *DB2 Administration Guide*.

## Capturing Event Monitor Data

DB2 event monitors can be created to automatically capture database activity. Activities can include database deadlocks, SQL statements, connections, transactions, tables, and buffer pools. The event monitor output is usually stored in files and analyzed using the DB2 Event Analyzer tool. The most common use for event monitors is to determine when and why deadlocks are occurring in the database system.

### Data Fragmentation

Table fragmentation should be checked periodically. The command REORGCHK can be issued to perform statistics update and verify if any tables are candidates for reorganization (defragmentation).

## DB2 Configuration Parameters

### Registry Parameters

DB2 maintains a set of environmental settings within the DB2 registry. The command **`db2set`** is used to change any of these settings and the Siebel Installation Guide provides recommended values for a number of parameters. For example, hash joins are currently enabled using the parameter **`DB2_HASH_JOIN=YES`**. The Siebel Installation Guide documents all required settings within the DB2 registry. Other registry parameters that should be set include: **`DB2_RR_TO_RS=YES`** and **`DB2_CORRELATED_PREDICATES=YES`**. These parameters are used to enable row level locking and assist the optimizer in plan selection. The instance needs to be stopped and restarted for these parameters to become effective.

### DBM Configuration Parameters

Instance level parameters are set using the command **`db2 set dbm cfg using <parm_name> <parm_value>.`** The Siebel Installation Guide documents all of the recommended settings. A full description of these parameters is provided in the *DB2 Administration Guide*. One such parameter is described here involves the use of server memory for sort operations. Ideally sort activity should be avoided through the use of indexes, but occasionally sorts are desirable as the overhead of index maintenance is unwarranted.

**SHEAPTHRES**

The total number of in-memory (piped) sorts is controlled by the combination of sort heap threshold (**SHEAPTHRES**) parameter (DBM configuration) and the **SORTHEAP** parameter (database configuration). A single sort heap will be dynamically allocated byDB2 on behalf of the application when it is required. If the sort operation can be performed within this memory area, then it is considered a *piped sort*. If the memory area is insufficient then a piped sort overflow may occur. These sort overflows can be easily monitored using theDB2 snapshot monitoring features.

### Database Configuration Parameters

A DB2 database consists of tables, indexes, transaction log files and a configuration file. The database configuration file is used to establish resources shared by all applications that access the database. Some of these shared resources include: sort buffers, buffer pools, and locks.

## Feedback

Any feedback in relation to the content of this document can be directed to its author Grant Hutchison at *ghutchis@ca.ibm.com*.

## Trademarks

The following terms are trademarks or registered trademarks of the IBM Corporation in the United States and/or other countries:

AIX
AIXwindows
AS/400
C Set++
C/370
DATABASE 2
DataJoiner
DataPropagator
DataRefresher
DB2
DB2 Connect
DB2 Universal Database
DB2 OLAP Server
Distributed Relational Database Architecture
DRDA
IBM
IMS
MVS/ESA
MVS/XA
OS/400
OS/390
OS/2
RISC System/6000
SQL/DS
SQL/400
S/370
System/370
System/390
VisualAge
VM/ESA
VSE/ESA
VTAM
WIN-OS/2

The following terms are trademarks or registered trademarks of the companies listed:
HP-UX is a trademark of Hewlett-Packard.
Java and HotJava are trademarks of Sun Microsystems, Inc.
Microsoft, Windows, Windows NT, Windows 95 and the Windows 98 are trademarks or registered trademarks of Microsoft Corporation.
Solaris is a trademark of Sun Microsystems, Inc.
UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.
Siebel, and/or other Siebel products referenced herein are trademarks of Siebel Systems, Inc., and may be registered in certain jurisdictions.
Other trademarks and trade names mentioned herein are the property of their respective owners.