Delphi Client / Server Suite 2.0
Client / Server Database Architecture

Lance Devin
Sr. Product Manager, Delphi

Borland International
7/2/2025

I. EXECUTIVE SUMMARY..

II. OVERVIEW...

III. CLIENT / SERVER FUNCTIONALITIES...

A. DELPHI CLIENT / SERVER SUITE 2.0 DATABASE APPLICATION ARCHITECTURE..
1. Benefits:..
2. Implementation: Data Modules, Scaleable Data Dictionary, Object Repository, and Visual Form Inheritance..
B. SQL MONITOR..
C. SQL EXPLORER..
D. INTERBASE NT - RELATIONAL DATABASE..

IV. DELPHI DATABASE COMPONENT ARCHITECTURE...

A. CONNECTIVITY AND TRANSACTIONS..
B. TABLES, STORED PROCEDURE AND QUERIES...
1. Filters:...
2. Update Mode:..
C. DATABASE CONTROLS...
1. Multi-Object Grid..
2. Enhanced DataGrid with Codeless Lookups..
3. DataListBox and DataComboBox...
D. REPORTSMITH...
E. QUICKREPORT...
F. IMPLEMENTATION...

V. BORLAND DATABASE ENGINE (BDE) ARCHITECTURE...

A. BORLAND DATABASE ENGINE COMPONENTIZED ARCHITECTURE..
B. HIGH PERFORMANCE NATIVE DRIVERS...
C. TRANSACTION MODELS...
1. Navigational Updates...
2. New Cached Updates..
D. TRANSACTIONS AND DELPHI EXCEPTION HANDLING...
E. HETEROGENEOUS DATABASE ACCESS...
F. 32-BIT WINDOWS SUPPORT...

VI. CONCLUSION...

I Executive Summary

In a short amount of time, hundreds of thousands of customers have used Delphi Client / Server to solve their business needs; including such organizations as Alcatel, American Cyanamid, American Stores, Arthur Anderson, AT&T, BMW, BP Shipping, Bank of America, BBC Television, British Telecom, City of Los Angeles, Compaq, Conoco, Coopers & Lybrand, DHL, Dover Elevators, EDS, Ernst & Young, Fiat, First National Bank of Chicago, Glaxo, KPMG, Mercury Communications, Netscape, Sarah Lee Knitting, Standard & Poors, SwissBank SG Warburg, Union Bank, US Marine Corps and many others.

The Banking, Financial, Chemical, Consulting, Communication, and Multimedia industries are using Delphi Client / Server for a broad range of applications that demand rapid development and high performance. Sample applications include:

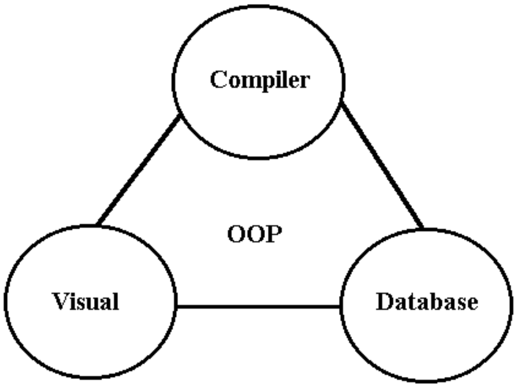
- A leading petrochemical company has created an ***executive information system*** to provide key operating and financial information on an hourly basis to over 600 users worldwide.
- A commercial bank has created a ***global funds transfer system*** supporting 25 currencies with secure transactions between major financial centers worldwide.
- A marketing research agency has created an ***executive information system*** to provide on-line analysis of sales data stored on an Oracle server.
- A rental company uses Delphi Client/Server to create an ***business automation application*** that tracks equipment nationwide across several locations with more than forty simultaneous users connecting to an InterBase server running on Windows NT.
- A major textile manufacturer has used Delphi Client/Server to develop an all new IS infrastructure for all ***accounting, manufacturing and sales information*** that will access an Oracle database with over 100 million rows.
- An Internet consulting group used Delphi Client/Server to create an application being used by advertising agencies, retailers and banks to ***manage pages on the World Wide Web***.
- A major industrial commodity markets analyst has created a ***global client/server system*** using Delphi Client/Server to access a 200 gigabyte Oracle database to produce up-to-date daily reports that will be deployed to thousands of customer worldwide.

Delphi Client / Server Suite incorporates integrated technologies that help solve your business needs. The new Database Application Architecture, SQL Monitor, Scaleable Data Dictionary, and Optimizing 32-bit Native Code Compiler, increase performance and reusability. Whether your applications are On-Line Transaction Processing Systems, Executive Information Systems, Decision Support Systems or Business Automation Systems, Delphi Client / Server Suite 2.0's object oriented and high performing architecture will allow you to produce solutions quickly.

In this paper we will discuss the solid architecture underlying Delphi Client / Server Suite 2.0. We will present some business and development problems and then identify technologies implemented in Delphi that quickly and simply solve them. The implementation and benefits of the SQL Explorer, Object Repository, Filters, Cached Updates, Borland Database Engine, Data Modules, and more will be presented in a framework that describes the Delphi application architecture.

II Overview

Delphi Client/Server Suite 2.0 is architected for high performance Client / Server Windows application development. Since the introduction in February 1995, Borland's Delphi Client/Server development tool has set a new standard in high-performance rapid application development. Delphi has won dozens of awards worldwide and has become the fastest growing visual tool. Delphi Client / Server Suite 2.0 combines an object oriented architecture, visual development and a native code compiler with Client/Server scalability to create robust business solutions.



Delphi Solutions for high performance applications: Delphi's optimizing 32 bit native code compiler is the fastest way to the fastest applications. Your applications result in royalty free, standalone executable files for rapid deployment. The new optimized linker makes EXEs up to 20-25% smaller and 300-400% faster than Delphi 1.0. Delphi Client / Server Suite 2.0 creates applications that run up to 15-20 times faster than interpreted p-code applications and in some cases up to 800 times faster than PowerBuilder.

Delphi Solutions for reusable and maintainable application objects: Object-oriented programming shortens development cycles by offering maximum reusability and maintainability. Delphi is a completely object oriented programming paradigm with support for polymorphism, encapsulation and inheritance. Delphi's new *Object Repository* stores and manages application objects for reuse, including: Forms, Data Modules, Experts, and DLLs for reuse. It centrally locates corporate assets so that they may be leveraged by the team to eliminate redundant development efforts.

Delphi Solutions for database speed and scalability: Delphi uses the high performance Borland Database Engine with SQL Links, 32-bit native drivers, to develop applications for DB2, Sybase - DBLib, Sybase CTLib, Microsoft SQL Server, Informix, Oracle, and InterBase. Delphi also has ODBC connectivity to access any ODBC compliant server such as Access, VSAM, or AS400. Delphi's high speed native access to database servers means the highest performing Client / Server applications. Delphi Client/Server Suite 2.0 also includes a 2 user development license of Interbase NT, a fully scaleable high performance SQL Relation Database. Regardless of where your data resides or the volume of your data, your application can quickly and easily get to it.

Delphi has added *Cached Update* transactions to offer a fast and flexible database interaction that is good for OLTP applications as well as high performance client/server applications. The cached update functionality will buffer a series of database inserts, deletes and modifications for processing on demand. By not updating the underlying tables directly, record locks and contentions are minimized resulting in shorter wait times by end-users.

Delphi Solutions for RAD and OOP database application development: Delphi Client / Server Suite 2.0 introduces a new *Database Application Architecture* where Rapid Application Design and Object-Oriented methodologies are applied to both the GUI and Database Business Logic. For the first time, *Visual Form Inheritance* allows any programmer to take advantage of Object Oriented techniques resulting in easier to maintain code and maximum reusability.

Delphi Solutions for Team Development: Delphi includes source code control with Intersolv's *PVCS Version Manager* to allow a team of developers to work most efficiently. Check In/ Check Out capabilities supplemented with Visual Differencing, Reporting, and Archive management assists in the development of deployable applications. PVCS, the industry leading source code control and configuration management software, helps large teams of developers work together and reduce development "chaos." This in turn reduces errors and brings product to market more quickly and more profitably.

The PVCS interface was built using Delphi's *Open Tools API* which allows for seamless integration with a variety of third party tools: CASE, Object Oriented Analysis and Design, Testing, Transaction Process Monitors, etc. Delphi integrates with the latest leading edge technologies, and ‘tried-and-true’ software development tools to provide developers with an environment with unparalleled productivity in PC software development.

To further enhance team development, the *Object Repository* centrally locates application objects / classes for easier distribution to the team. All objects can be accessed by reference, by inheritance or by copy. Application objects include the new *Data Module* which acts as your applications information core for defining data access and business rules. The Data Module logically partitions applications into Visualization, Business Logic, and Data Access.

Delphi Solutions for consistent database access: Delphi Client Server Suite 2.0 delivers a *Scaleable Data Dictionary* that enables consistent database access. Field attributes (e.g. min, max, default value, etc.) are defined once and reused by applications to enforce data integrity. This ensures that each application behaves according to the data-validation rules defined at a meta-layer for the database. The dictionary is surfaced by the *SQL Explorer*, an integrated tool that modifies databases and extended field attributes.

Delphi Solutions for robust applications: Delphi was built from the ground up to allow for the creation of robust, fault tolerant applications. Delphi marries strong language constructs such as *Exception Handling* with a object oriented database architecture to provide simple and robust transaction control. Developers can create and raise their own exceptions to further protect against errors and provide new mechanisms for recovery and stability.

A Complete Suite of Tools:

Delphi Client / Server Suite 2.0 includes a complete suite of tools for client side development and server management with exclusive lools like SQL Explorer, SQL Monitor and InterBase NT.

III Client / Server Functionalities

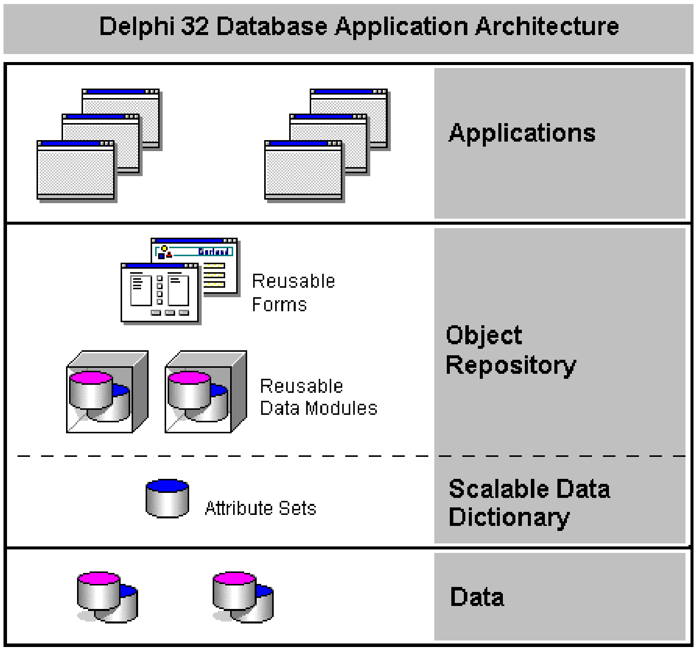
Delphi Client / Server Suite 2.0 includes an integrated suite of tools for building high performance Client / Server applications with such extensive features as:

- *Object Oriented Database Application Architecture*
- *Flexible Client / Server transaction models.*
- *Centralized Object Repository for team development*
- *SQL Monitor for SQL Testing and Tuning*
- *SQL Explorer for integrated administration of database servers*
- *Data Module Objects for separating business logic from visual data representation*
- *Scaleable Database Dictionary for consistent use of extended field attributes*

Delphi Client / Server Suite 2.0 introduces an object oriented architecture to many new Client / Server features. The **Data Module**, **Scaleable Database Dictionary** and **Object Repository** are technologies that become the framework of a unique **Database Application Architecture**. This architecture then becomes the basis for building high performance Client / Server applications. Details on the architecture, transaction models and SQL capabilities in Delphi Client / Server Suite 2.0 are described below.

A Delphi Client / Server Suite 2.0 Database Application Architecture

One objective of Client / Server application design is to develop a reusable set of Objects, Business Rules and Forms. Delphi Client / Server Suite 2.0 uniquely implements an architecture incorporating Rapid Application Development and Object Oriented Design to reduce development time and improve maintainability. Additionally, this architecture allows you to separate GUI, Business Logic and Database design with the following benefits.



Delphi Client / Server Suite 2.0's architecture coordinates creation of reusable forms, objects, and business logic.

1 Benefits:

1. Separation of the GUI Design from the data and its associated logic means that changes in either of these areas does not impact the successful usage of the other. Changes in either the GUI or business logic can be implemented according to their own requirements.

2. Separation of the GUI Design from the data and its associated logic allows developers to apply their skill sets appropriately. No longer does a single developer have to have expertise in Database Design, Business Methodologies and GUI Design to create an effective Client / Server application. Application development becomes parallelized resulting in faster development.

3. Visual Form Inheritance allows object modules to be codelessly reused which reduces development time and eases maintenance. As business logic changes or corporate interface standards evolve, changes will automatically cascade to all inherited objects.

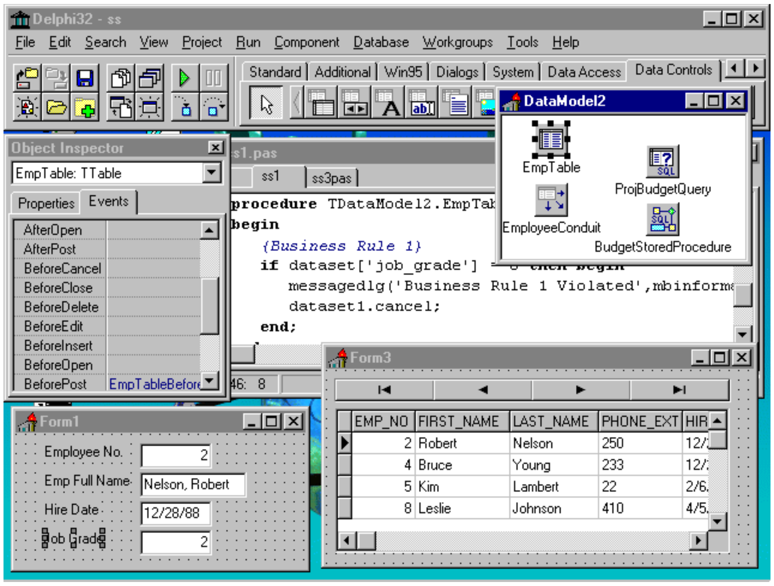
4. The Object Repository stores reusable forms and Data Modules, that can be shared by a team of developers to eliminate duplicate coding and design work. This improves developer productivity.

2 Implementation:

Data Modules, Scaleable Data Dictionary, Object Repository, and Visual Form Inheritance

a Data Module Objects

Delphi Client / Server Suite 2.0's Data Module Objects act as your applications information core by providing a designated central location for defining data access and business rules. The Data Module Object separates business logic from the GUI and acts as a codeless way to connect and manage this business logic from a single location.



EmpTable: TTable

AfterOpen	
AfterPost	
BeforeCancel	
BeforeClose	
BeforeDelete	
BeforeEdit	
BeforeInsert	
BeforeOpen	
BeforePost	EmpTableBefore

```
procedure TDataModel2.EmpTak
begin
  {Business Rule 1}
  if dataset['job_grade'] 
    messagedlg('Business Rule 1 Violated',mbinform
    dataset1.cancel;
  end;
```

EMP_NO	FIRST_NAME	LAST_NAME	PHONE_EXT	HIR
2	Robert	Nelson	250	12/
4	Bruce	Young	233	12/
5	Kim	Lambert	22	2/6.
8	Leslie	Johnson	410	4/5.

Employee No. 2

Emp Full Name Nelson, Robert

Hire Date 12/28/88

Job Grade 2

Delphi Client / Server Suite 2.0's new Data Module Objects encapsulate Business Rules.

- Business logic can be applied to Tables, Stored Procedures, and Queries by creating methods on Before and After events such as posts, deletes, inserts and edits. This allows you to create new business objects more easily.

- Master Detail Data relationships are defined in a codeless manner in Delphi. The Developer can create Client / Server applications easily and quickly by visually setting properties on the Datasources or by using the Database Form Expert.

- Applications and forms can be visually linked to the Data Module to propagate business logic without writing extra code.

    - Data Modules are classes of objects that pertain to the interaction of data with the database server. Isolating all database operations in Client / Server application makes application maintenance simple.

Logical Application Distribution:
The Data Module allows developers to create a logical n-tier application environment by centrally locating business logic and separating it from the GUI and database design. This is a solid foundation for physical support of distributed applications. Support for physical n-tier application and server architectures is currently available for Delphi through third party libraries such as IBM's CICS, ATT's TopEnd, Novell's Tuxedo, Digital's Object Broker, Orbix's IONA and others.

Network OLE and VB4's Remote Automation technology:
In the Windows environment, Microsoft operating system technologies such as Visual Basic 4.0's remote automation and the forthcoming Network OLE can be used to distribute application logic among servers. Because Delphi Client / Server Suite 2.0 fully supports these technologies and can create high performance OLE automation controllers and servers, Delphi will allow developers to create partitioned applications easily.

b Scaleable Database Dictionary

The database dictionary stores and uses customized information about the contents of the data in your tables. The dictionary ties the familiar model of working with fields through the property inspector to a permanent storage facility. The data dictionary holds information about extended field attributes like min, max, default values, and display preferences. Using the data dictionary has two advantages:

1. Consistency: common field attributes are centrally stored in the data dictionary and then retrieved at application design time. A developer can create extended field attribute domains and the team can appropriately apply them to fields.

2. Network Traffic Reduction: Delphi allows for data validation at the client or at the server. The data dictionary can maintain extended field attributes that perform client side data validation efficiently by reducing network traffic.

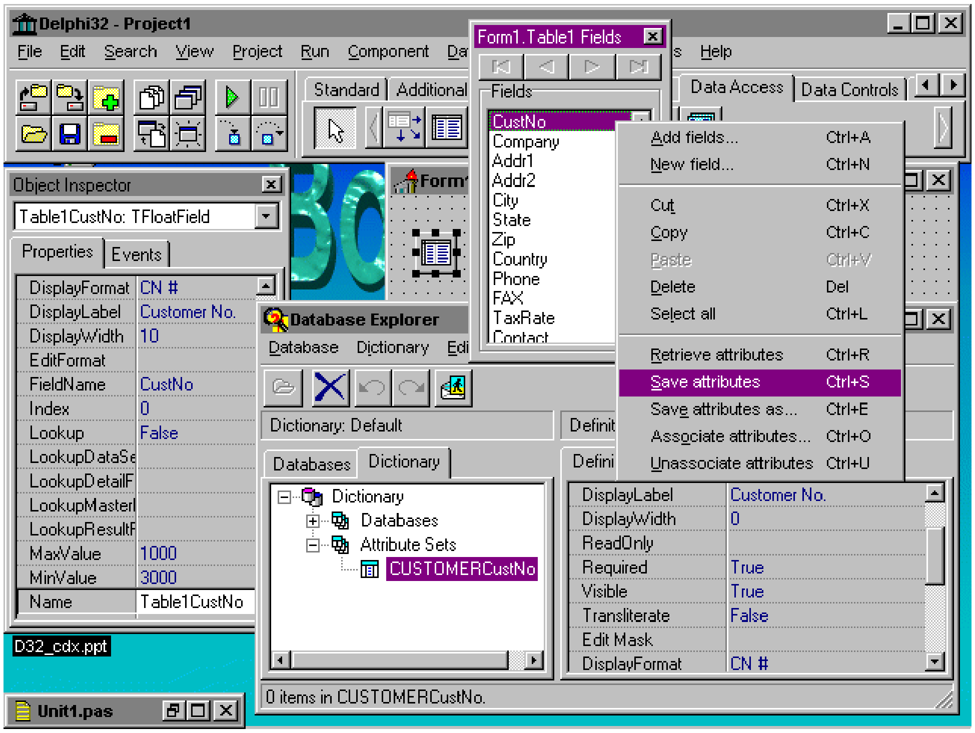


Table1CustNo: TFloatField

DisplayFormat	CN #
DisplayLabel	Customer No.
DisplayWidth	10
EditFormat	
FieldName	CustNo
Index	0
Lookup	False
LookupDataSe	
LookupDetailF	
LookupMasterl	
LookupResultF	
MaxValue	1000
MinValue	3000
Name	Table1CustNo

Dictionary: Default

DisplayLabel	Customer No.
DisplayWidth	0
ReadOnly	
Required	True
Visible	True
Transliterate	False
Edit Mask	
DisplayFormat	CN #

0 items in CUSTOMERCustNo.

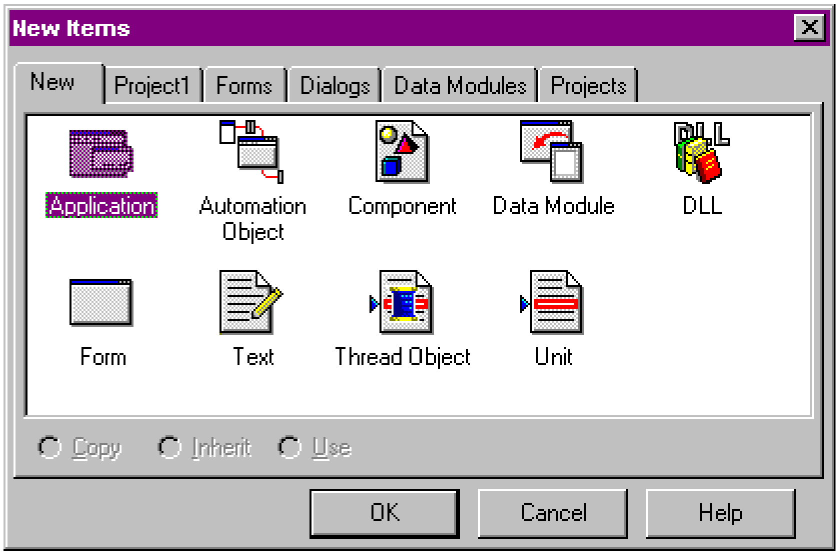
The Scaleable Data Dictionary maintains consistency by storing attribute sets for Database Schema.

c *Object Repository*

Delphi Client / Server Suite 2.0's Object Repository stores and manages application objects: Forms, Data Modules, Experts, and DLLs. In essence, it centrally locates corporate assets so that they may be leveraged by the team to eliminate redundant development efforts. As objects proliferate, the repository increases in importance.

The Object Repository:

- Supports team development practices by referencing objects on a network.
- Is customizable so that developers can define their own logical groupings of objects to facilitate reuse.

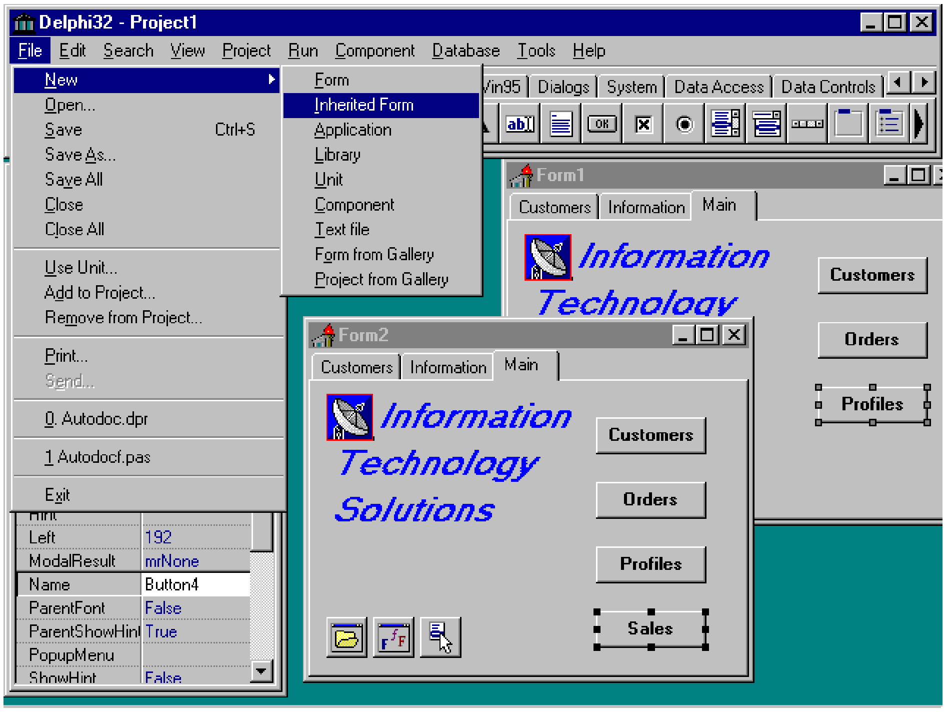


Object Repository centrally locates application objects and enhances team development.

d *Visual Form Inheritance and Form Linking*

Developing a corporate standard in applications is important. Ensuring that these standards are adhered to is more difficult. Visual Form Inheritance and Form Linking extends object oriented programming to a visual paradigm ensuring that corporate and programmatic standards are maintained from project to project. In conjunction with the Object Repository, these standards are centrally managed resulting in faster project turn-around time.

Visual Form Inheritance allows anyone to take advantage of object-oriented reusability and maintainability by providing a codeless way to use inheritance, encapsulation and polymorphism.



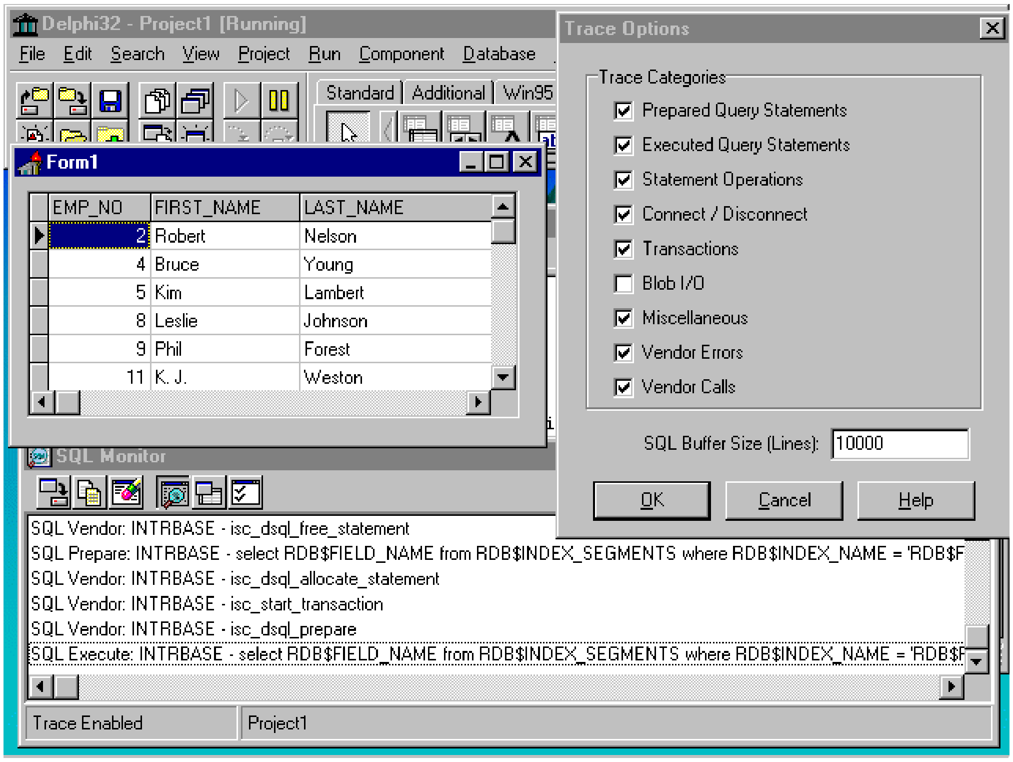
Hint	
Left	192
ModalResult	mrNone
Name	Button4
ParentFont	False
ParentShowHint	True
PopupMenu	
ShowHint	False

Visual Form Inheritance simplifies Object Oriented reusability.

B SQL Monitor

Delphi Client / Server Suite 2.0 is the only RAD tool that integrates a native SQL monitor for testing, debugging and tuning SQL queries in Client / Server applications. This in turn increases developer productivity and application performance.

The SQL Monitor enables the developer to trace calls between the client and server. This information allows the developer to find problematic SQL statements and then optimize the SQL transactions. A series of trace options lets the developer customize the amount and type of information that is reported on. The SQL Monitor helps the developer know that the SQL in the application is being performed optimally, what is the SQL generated by the Borland Database Engine, if the Database Client Libraries are functioning properly, and if the database server is executing a run-away query. The additional capabilities to save and print the session log enables more thorough testing.



EMP_NO	FIRST_NAME	LAST_NAME
2	Robert	Nelson
4	Bruce	Young
5	Kim	Lambert
8	Leslie	Johnson
9	Phil	Forest
11	K. J.	Weston

Trace Options

Trace Categories

- ☑ Prepared Query Statements
- ☑ Executed Query Statements
- ☑ Statement Operations
- ☑ Connect / Disconnect
- ☑ Transactions
- ☐ Blob I/O
- ☑ Miscellaneous
- ☑ Vendor Errors
- ☑ Vendor Calls

SQL Buffer Size (Lines): 10000

```
SQL Vendor: INTRBASE - isc_dsql_free_statement
SQL Prepare: INTRBASE - select RDB$FIELD_NAME from RDB$INDEX_SEGMENTS where RDB$INDEX_NAME = 'RDB$F
SQL Vendor: INTRBASE - isc_dsql_allocate_statement
SQL Vendor: INTRBASE - isc_start_transaction
SQL Vendor: INTRBASE - isc_dsql_prepare
SQL Execute: INTRBASE - select RDB$FIELD_NAME from RDB$INDEX_SEGMENTS where RDB$INDEX_NAME = 'RDB$F
```

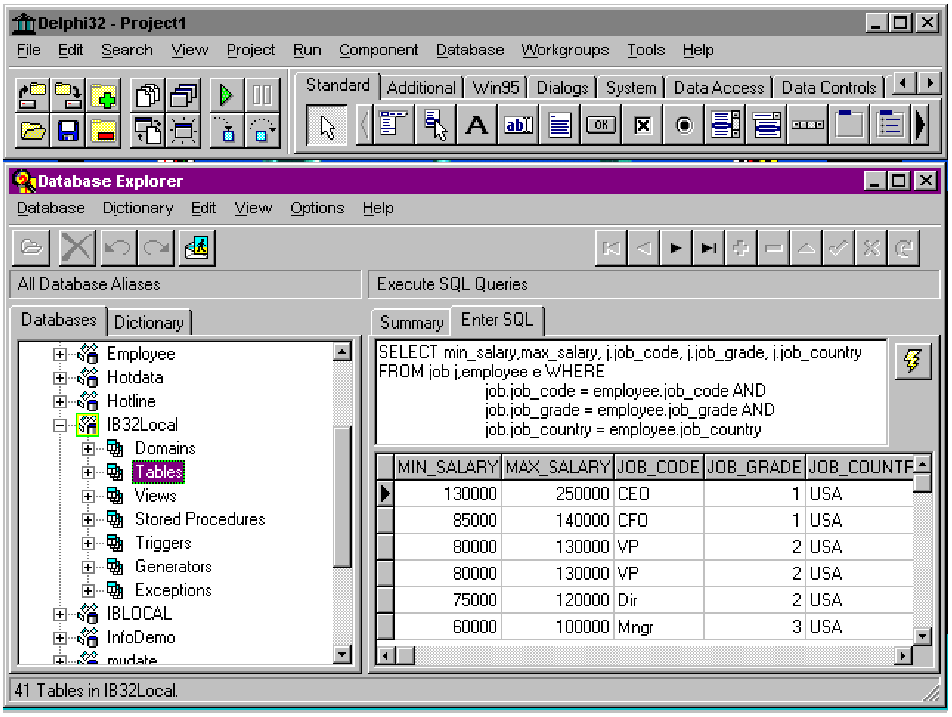
Trace Enabled | Project1

Delphi Client / Server Suite 2.0 is the only RAD tool that integrates a native SQL monitor for testing, debugging and tuning SQL in Client / Server applications.

C SQL Explorer

The SQL Explorer provides the information center for your database management demands; it supports the creation and modification of tables, aliases, stored procedures, triggers and business rules through interactive SQL. This graphical tool is an integrated database schema and content management utility tailored to the needs of professional database developers.

The SQL Explorer, unique to Delphi, makes database administration easier and more intuitive than having to use a separate non-integrated tool. A simple to use graphical interface is a perfect way to represent the complex relationships that exist in a database server. The SQL Explorer presents schema information from Oracle, Sybase, InterBase, Informix, DB2 and others. The developer can drag and drop fields, tables, and stored procedures onto the Delphi application form to build Client / Server database applications quickly. The developer can also issue SQL statements directed to multiple servers and multiple databases.



```
SELECT min_salary,max_salary, j.job_code, j.job_grade, j.job_country
FROM job j,employee e WHERE
              job.job_code = employee.job_code AND
              job.job_grade = employee.job_grade AND
              job.job_country = employee.job_country
```

MIN_SALARY	MAX_SALARY	JOB_CODE	JOB_GRADE	JOB_COUNTR
130000	250000	CEO	1	USA
85000	140000	CFO	1	USA
80000	130000	VP	2	USA
80000	130000	VP	2	USA
75000	120000	Dir	2	USA
60000	100000	Mngr	3	USA

41 Tables in IB32Local.

The SQL Explorer is an integrated tool for administering SQL and PC Databases from within the Delphi Client / Server Suite 2.0 environment.

The SQL Explorer, also manages the Scaleable Database Dictionary. The simple to use interface enables the developer to easily define new domains of extended field attributes and then associate those to a field. The next time the field is used in an application, all the attributes are automatically applied.

D *InterBase NT - Relational Database*

Delphi Client / Server Suite 2.0 includes a two user InterBase NT developer license. Developers can create standalone client / server applications using a scaleable relational database. When the volume of data, or size of application grows, both the InterBase relational database and the Delphi application will scale accordingly.

InterBase is Borland's high performance, cross platform SQL Server. InterBase is available on over 15 operating systems, including: Windows 3.1, Windows 95, Windows NT, NetWare, SCO, Sun OS, Sun Solaris, HP-UX, IBM AIX, SGI IRIX, etc. InterBase is ANSI SQL 92 entry level conformant, supports server events for event driven programming, and has an exceptional concurrency model for multiuser access. InterBase offers record level locking and due to its Multi-Generational Architecture delivers superior performance because database read operations do not block database write operations.

Local InterBase, also available in Client / Server Suite 2.0, provides Delphi developers with their own single user ANSI 92 SQL conformant server for prototyping and development of true client/server applications on Windows 95 or Windows NT. Local InterBase has all of the same functionality as the multiuser versions of InterBase available for NT and Unix, including transaction control, stored procedures, triggers, and even event alerters, which enable event driven programming. This means that development can occur on a laptop while on the train, airplane, or at the customer site, and that the final database to be used can be changed when the application is ready to be deployed.

Using Delphi Client/Server Suite 2.0, developers can design, prototype, and test their Delphi/InterBase applications on one machine. InterBase offers an exceptional Windows 95 GUI interface including configurable property sheets, native 32 bit tools like the Server Manager and Interactive SQL tools, and the complete documentation in Windows 95 Help.

InterBase ensure that data is always available due to their excellent multiuser performance, high security, and fast recovery features. InterBase is used in the aeronautical industries by companies like Boeing and Lockheed for manufacturing, by the Money Store and many other banking institutions, and in financial trading centers like the Philadelphia, Boston, and Russian Stock Exchanges. The common thread among all these customers is the need for excellent multiuser performance, high security, and fast recovery when system failures occur.

IV Delphi Database Component Architecture

Summary: Delphi Database Component Architecture offers modular and complete control of:

- *Transactions and database connectivity*
- *Queries, Stored Procedures, Tables and Result Sets*
- *User Interaction Models*
- *Information Reporting*

Delphi's Database Component Architecture is a high level object oriented encapsulation of the Borland Database Engine Architecture. The object oriented architecture offers a simple application programming interface with three benefits.

1. Database developers concentrate on getting the data models correct while the GUI design engineers concentrate on the application front end.

2. Developers can concentrate on fulfilling a business need instead of having to build everything themselves. Developers can purchase components from third party vendors to extend Delphi functionality.

3. The amount of time it takes to learn how to use Delphi database controls is significantly reduced in comparison to many other tools because the GUI design model is the same as the database design model.

A Connectivity and Transactions

The Database Component allows the developer to programatically change the type of connection (SQL Pass Through Mode) that is maintained between an application and a database server. The ability to change the type of connection means that the complete power of the database server can be harnessed to improve the flexibility of an application. It also means that connections can be managed to reduce the number of user accesses seen by the database server resulting in hard cost savings.

The Database component can also maintain a persistent connection to a server so that a server connection operation does not have to be re-established each time. This speeds up applications.

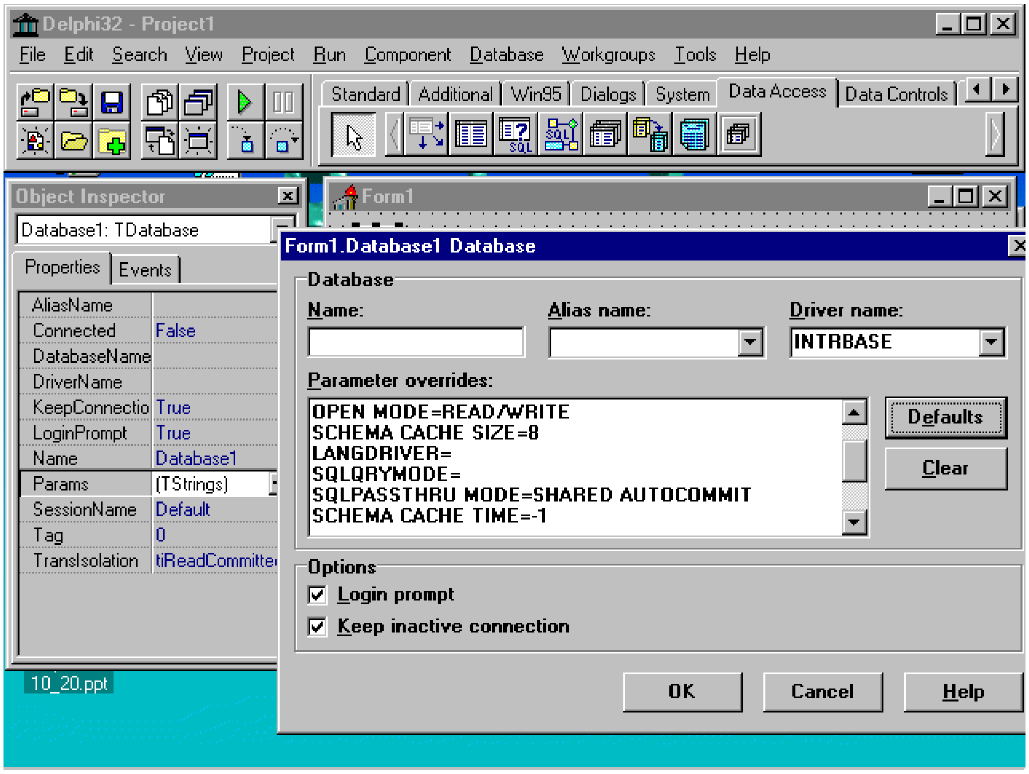
There exist three different types of connectivity maintained by Delphi:

1. **Shared AutoCommit:** The Borland Database Engine implements a navigation update methodology. Singleton transactions are created to maintain the strictest level of coordination between the client result set and the data in the underlying database tables.

2. **Shared NoAutoCommit:** This mode means that the Borland Database Engine is aware of all SQL transactions on the server and will coordinate the client application result sets with the underlying data on the database server. However, commits of the singleton transactions are not automatic and must fall under programmatic control.

3. **Not Shared:** The Borland Database Engine in this mode doesn't know what transactions are taking place on the server and acts merely as a conduit of SQL statements to a back-end. This means there is no coordination between the client result sets and the underlying database tables. The developer must refresh the client view into the data. What this offers is access to non-standard SQL statements supported by the individual database servers. It also allows for run-time maintenance of the database server from within an application.

The flexibility offered by programatically choosing the type of connection means that the developer can chose to optimize for performance, for minimal contentions, for cost and/or for the amount of server control.


Delphi32 - Project1
File Edit Search View Project Run Component Database Workgroups Tools Help
Standard Additional Win95 Dialogs System Data Access Data Controls
Object Inspector
Database1: TDatabase
Properties Events
Form1
Form1.Database1 Database
10_20.ppt
Defaults
Clear
OK
Cancel
Help

AliasName	
Connected	False
DatabaseName	
DriverName	
KeepConnectio	True
LoginPrompt	True
Name	Database1
Params	(TStrings)
SessionName	Default
Tag	0
TransIsolation	tiReadCommitte

Database

Name:

Alias name:

Driver name: INTRBASE

Parameter overrides:

```
OPEN MODE=READ/WRITE
SCHEMA CACHE SIZE=8
LANGDRIVER=
SQLQRYMODE=
SQLPASSTHRU MODE=SHARED AUTOCOMMIT
SCHEMA CACHE TIME=-1
```

Options

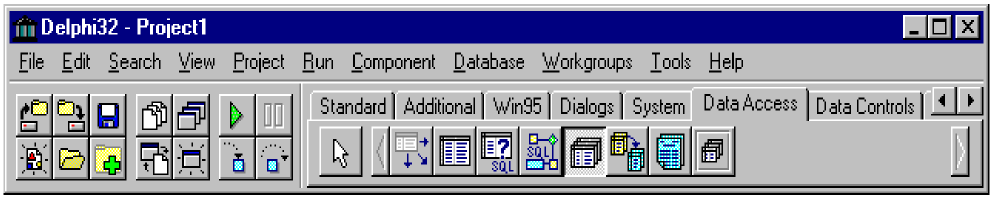
☑ Login prompt

☑ Keep inactive connection

Property editors let you easily set Complex database properties

B Tables, Stored Procedure and Queries

Tables, Stored Procedures and Queries are the components of Client / Server applications. All three of these object types share certain characteristics and are therefore implemented from a common class known as a Dataset. The separation of the Dataset from connectivity and data visualization means that implementation of business rules does not impact any other areas.


Delphi32 - Project1
File Edit Search View Project Run Component Database Workgroups Tools Help
Standard Additional Win95 Dialogs System Data Access Data Controls
SQL

Delphi Client / Server Suite 2.0 componentizes Client / Server database access for drag and drop creation of applications.

Because of the componentization of the Borland Database architecture business logic can be applied to Tables, Stored Procedures, and Queries by creating methods on Before and/or After events such as posts, deletes, inserts and edits. This allows you to create new objects for flexibility.

1 Filters:

After a general query of a database it is common to want to successively pare down the list of results or to move through the list based on further criteria. Filters offer a flexible mechanism for subsetting the result set either on the client or on the server. In this way, the developer can choose what will offer the highest performance with the most flexibility.

Filter expressions are easily written in the Object Pascal language and have no limitations that are inherent in 4GL languages with respect to scope and breadth of function creation.

Filters facilitate the way people work by allowing users to scan through data more effectively. Being able to drill into a result set is one way to turn data into information by mapping to the way people work.

2 Update Mode:

The UpdateMode property determines how Delphi will find records being updated in a SQL database. This property is important in a multi-user environment when multi-user updates can cause conflicts.

When a user posts an update, Delphi uses the original record values to find the record in the database. This approach is called optimistic locking. The Update Mode specifies which table columns Delphi uses to find the record. In SQL terms, Update Mode specifies which columns are included in the WHERE clause of an UPDATE statement. If Delphi cannot find a record with the original values in the columns specified (if another user has changed the values in the database), Delphi will not make the update and will generate an exception, so that the application can implement custom behavior, perhaps logging the results.

The Update Mode property may have the following values:

- WhereAll (the default): Delphi uses every column to find the record being updated. This is the most restrictive mode.

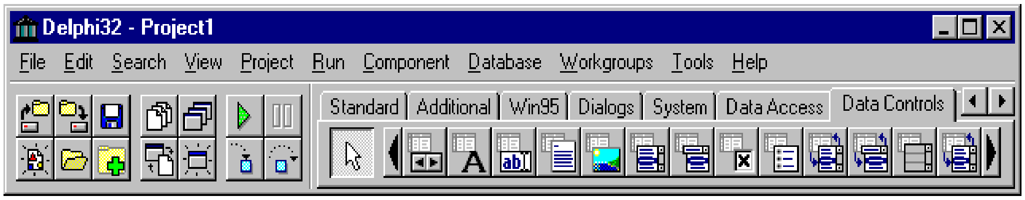
- WhereKeyOnly: Delphi uses only the key columns to find the record being updated. This is the least restrictive mode and should be used only if other users will not be changing the records being updated.

- WhereChanged: Delphi uses key columns and columns that have changed to find the record being updated.

Using the Object Inspector to visually change the Update Mode provides easy control for updating records in database servers. The Update Mode reduces the conflicts that arise in large, heavily used Client / Server applications which results in higher performing applications.

C Database Controls

Delphi Client / Server Suite 2.0 provides over 15 database visualization controls so that applications can take on an appealing and functional design. These controls are easy to use and because they are object oriented, completely extensible and customizable. Database controls such as Edits, Memos, and Database Grids can perform sophisticated Client / Server operations without requiring code. In addition, all database controls will take on the extended field attributes defined in the Scaleable Database Dictionary so that consistent user interface and data validation can be applied across applications.

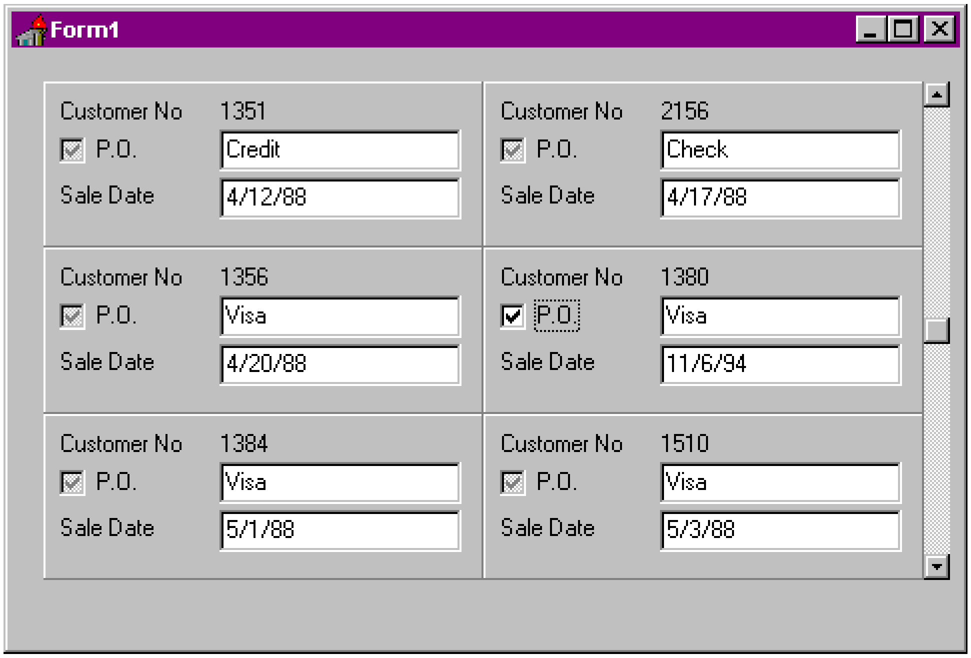

Delphi32 - Project1
File Edit Search View Project Run Component Database Workgroups Tools Help
Standard Additional Win95 Dialogs System Data Access Data Controls

Delphi Client / Server Suite 2.0 **has over 15 Database GUI Controls**

New to Delphi Client / Server Suite 2.0 are the Multi-Object Grid, an enhanced DataGrid with codeless lookup functionality, and enhanced database list and combo boxes. These controls offer the flexibility to convey information to the user in a way that corresponds to the business need and application requirements.

1 Multi-Object Grid

The multi-object grid allows the developer to place other database controls into a single row of this grid. CheckBoxes, Edits and so forth can be stacked or visually placed where you like, and with full replication. The flexibility and simplicity of conveying information in this way adds to the presentation of information and productivity of users.

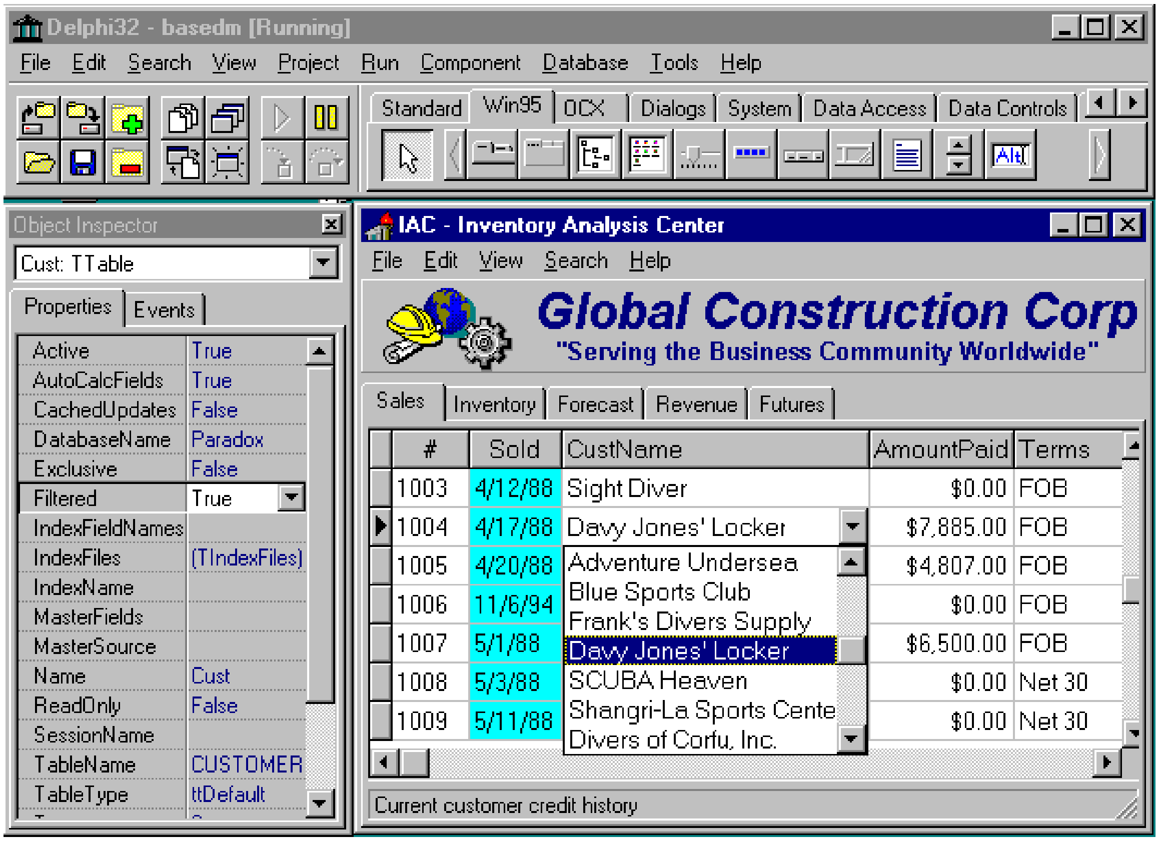


Customer No	1351	Customer No	2156
☑ P.O.	Credit	☑ P.O.	Check
Sale Date	4/12/88	Sale Date	4/17/88
Customer No	1356	Customer No	1380
☑ P.O.	Visa	☑ P.O.	Visa
Sale Date	4/20/88	Sale Date	11/6/94
Customer No	1384	Customer No	1510
☑ P.O.	Visa	☑ P.O.	Visa
Sale Date	5/1/88	Sale Date	5/3/88

Delphi Client / Server Suite 2.0's new Multi-Object Grid allows the developer to design a custom window of information

2 Enhanced DataGrid with Codeless Lookups

Applications like Executive Information Systems and Decision Support Systems require consistent data for accurate reporting of daily operations. The new DataGrid in Delphi Client / Server Suite 2.0 allows for codeless lookups between tables which ensures data validation and consistency. Drop down lists can now be used to supply information from one table to store in another. The developer can now create a data entry model that provides for consistent data entry.


Delphi32 - basedm [Running]
File Edit Search View Project Run Component Database Tools Help
Standard Win95 OCX Dialogs System Data Access Data Controls
Object Inspector
Properties Events
IAC - Inventory Analysis Center
File Edit View Search Help
Global Construction Corp
"Serving the Business Community Worldwide"
Sales Inventory Forecast Revenue Futures

Cust: TTable

Property	Value
Active	True
AutoCalcFields	True
CachedUpdates	False
DatabaseName	Paradox
Exclusive	False
Filtered	True
IndexFieldNames	
IndexFiles	(TIndexFiles)
IndexName	
MasterFields	
MasterSource	
Name	Cust
ReadOnly	False
SessionName	
TableName	CUSTOMER
TableType	ttDefault

#	Sold	CustName	AmountPaid	Terms
1003	4/12/88	Sight Diver	$0.00	FOB
1004	4/17/88	Davy Jones' Locker	$7,885.00	FOB
1005	4/20/88		$4,807.00	FOB
1006	11/6/94		$0.00	FOB
1007	5/1/88		$6,500.00	FOB
1008	5/3/88		$0.00	Net 30
1009	5/11/88		$0.00	Net 30

Adventure Undersea
Blue Sports Club
Frank's Divers Supply
Davy Jones' Locker
SCUBA Heaven
Shangri-La Sports Cente
Divers of Corfu, Inc.

Current customer credit history

Delphi Client / Server Suite 2.0's new codeless lookups in the enhanced datagrid.

The DataGrid has also been enhanced to be independent of the Dataset. This provides the flexibility to use different grids to look at the same table, query or stored procedure in different ways. The developer can apply column attributes such as position, fonts, colors, headers, width, etc. to highlight important information and to convey different messages about the data.

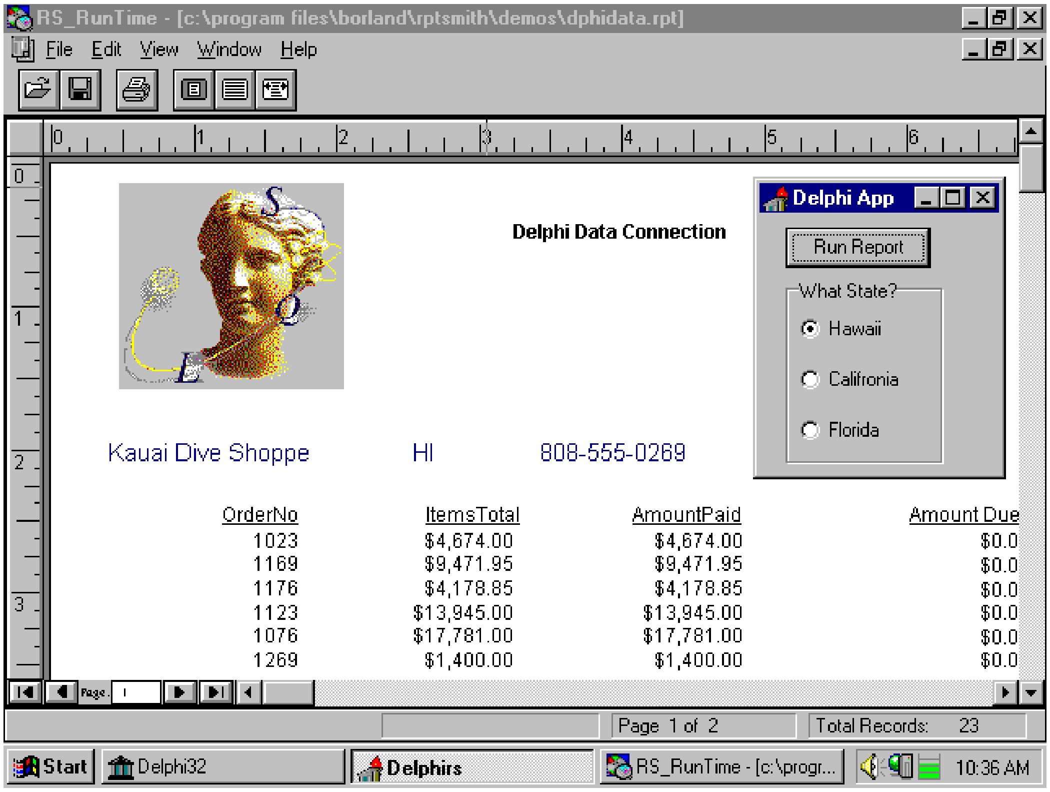
3 DataListBox and DataComboBox

The DataListBox and the DataComboBox seamlessly link information between multiple Datasources whether they be queries, stored procedures or tables. Developers can now codelessly order pick lists so that users can find data more easily and enhance their productivity. Client / Server applications rely on normalized, validated and coherent data so that important business decisions can be made accurately.

D ReportSmith

ReportSmith turns data into information that assists in making every day business decisions. ReportSmith can directly access the Delphi Datasets offering easy integration. This allows client/server developers to create complex database reports against extremely large quantities of data. ReportSmith supports columnar, crosstab, form, or mailing label reports, offering the flexibility to view data typically found in executive information systems and decision support systems.

One language can control all aspects of Client / Server applications reducing the learning curve of your developers. By sharing the Delphi data connection from within ReportSmith, developers are able to control the content, sort order, and grouping of a report with code.


RS_RunTime - [c:\program files\borland\rptsmith\demos\dphidata.rpt]
File Edit View Window Help
Delphi App
Run Report
Page 1
Start
Delphi32
Delphirs
RS_RunTime - [c:\progr...
10:36 AM

Delphi Data Connection

What State?
● Hawaii
○ Califronia
○ Florida

Kauai Dive Shoppe HI 808-555-0269

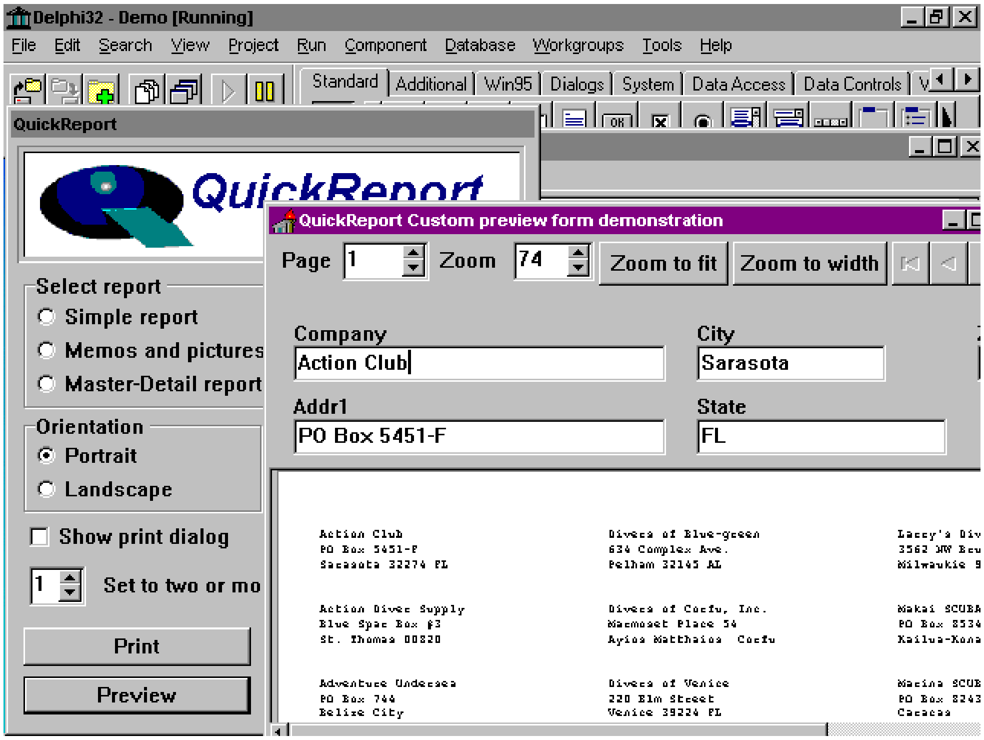
OrderNo	ItemsTotal	AmountPaid	Amount Due
1023	$4,674.00	$4,674.00	$0.0
1169	$9,471.95	$9,471.95	$0.0
1176	$4,178.85	$4,178.85	$0.0
1123	$13,945.00	$13,945.00	$0.0
1076	$17,781.00	$17,781.00	$0.0
1269	$1,400.00	$1,400.00	$0.0

Page 1 of 2 Total Records: 23

ReportSmith offers high volume Client / Server reporting.

E QuickReport

Delphi Client / Server Suite 2.0 also includes QuickReport, an integrated set of Delphi components that rapidly creates columnar, master-detail, and label reports. While Report Smith is optimized for large volumes of data that are typically found in corporate client/server applications, QuickReport is optimized for smaller volumes of data and simpler reporting styles.


Delphi32 - Demo [Running]
File Edit Search View Project Run Component Database Workgroups Tools Help
Standard Additional Win95 Dialogs System Data Access Data Controls
QuickReport
QuickReport
QuickReport Custom preview form demonstration
Zoom to fit
Zoom to width
Print
Preview

Select report
○ Simple report
○ Memos and pictures
○ Master-Detail report

Orientation
● Portrait
○ Landscape

☐ Show print dialog

1 Set to two or mo

Page 1 Zoom 74

Company: Action Club
City: Sarasota
Addr1: PO Box 5451-F
State: FL

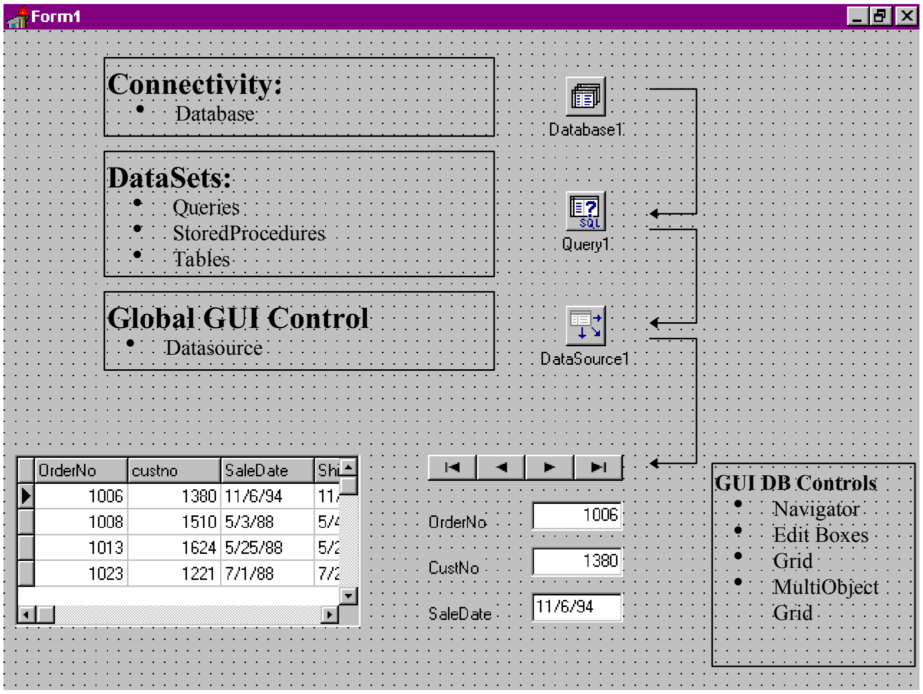
```
Action Club                 Divers of Blue-green          Lacey's Div
PO Box 5451-F               634 Complex Ave.              3562 NW Bru
Sarasota 32274 FL           Pelham 32145 AL               Milwaukie 9

Action Diver Supply         Divers of Corfu, Inc.         Makai SCUBA
Blue Spar Box #3            Marmoset Place 54             PO Box 8534
St. Thomas 00820            Ayios Matthaios  Corfu        Kailua-Kona

Adventure Undersea          Divers of Venice              Marina SCUB
PO Box 744                  220 Elm Street                PO Box 8243
Belize City                 Venice 39224 FL               Caracas
```

QuickReport integrates data reporting into Delphi Client / Server Suite 2.0 applications.

QuickReport is comprised of eleven components that intuitively allow the developer to create report bands, attach to Delphi datasets, and then finally lay out report fields. Printing, Previewing, and Page Orientation are invoked with simple object methods making it easy to incorporate reports into your client / server applications. QuickReport also makes it possible to preview reports in your application so that you can see the information as it would appear on paper. This allows you to look at the information in new ways.

F *Implementation*

The Database Component Architecture in Delphi Client / Server Suite 2.0 is an object oriented implementation of the Borland Database Engine. Each object has a modular and tight component API that manages a single aspect of Client / Server database applications. In comparison to a single object/window that manages all aspects of the application, Delphi's componentization offers a degree of isolation between the database and the GUI with the benefits of a shorter learning curve and easier maintenance.



Connectivity:
- Database

DataSets:
- Queries
- StoredProcedures
- Tables

Global GUI Control
- Datasource

OrderNo	custno	SaleDate	Shi
1006	1380	11/6/94	11/
1008	1510	5/3/88	5/4
1013	1624	5/25/88	5/2
1023	1221	7/1/88	7/2

OrderNo 1006

CustNo 1380

SaleDate 11/6/94

GUI DB Controls
- Navigator
- Edit Boxes
- Grid
- MultiObject Grid

Delphi's Component Architecture eases code maintenance.

- Session and Database components maintain information to manage transactions and changes to the database server for each connection.
- Datasets (e.g. query, stored procedure and table) allow the developer to control the communication between Delphi and the Borland Database Engine by providing over 15 events with which to interact.
- Datasources manage the interaction of the data record with all the GUI data controls.
- Over 15 data visualization (GUI) controls manage the presentation of data.

This discrete object model provides control of database operations. The programmer can apply scope to business rules and manage every aspect of the Client / Server application. Business rules can be applied at the GUI, the Data Record, the Table, the Database, the Database Server or any combination of the above to provide the most modular and flexible application.

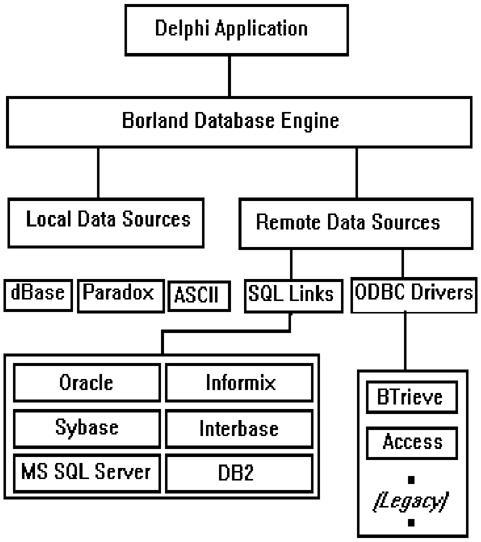
V Borland Database Engine (BDE) Architecture

Summary: The Borland Database Engine offers:

- *High Speed Native Drivers to Database Servers.*
- *A componentized architecture for robust applications.*
- *Flexible transaction models*
- *Heterogeneous database access*
- *Full 32-bit operating system support*

As the core engine, the Borland Database Engine provides unparalleled compatibility and efficiency in accessing data stored in ORACLE, Sybase, Informix, InterBase, DB2, and MS SQL Server using high speed SQL Links native drivers, from dBASE and Paradox data formats using a native engine, or from other formats using the ODBC standard interface. The Borland Database Engine is also the heart of Borland's own database products for the Windows environment: Paradox for Windows, and Visual dBASE. Borland Database Engine is also field proven: it has been in commercial applications since 1990 and tested by well over two million users. The Borland Database Engine is designed from the ground up to be fast, flexible, powerful, scaleable and extensible.

A Borland Database Engine Componentized Architecture



The Borland Database Engine Architecture

The new 32-bit Borland Database Engine consists of the following components that will allow for the fastest Client / Server applications and application scalability:

- **A common database query engine** that supports Structured Query Language (SQL). The Borland Database Engine has been optimized to efficiently deal with the semantics of both set-oriented SQL and record-oriented navigational databases through a single and unified engine. As a result, customer applications can fully benefit from the strengths of both access modes while offering performance and scalability. The new 32-bit query engine has full SQL 92 compliance and offers new functionality such as transactional support for local databases.

- **High Performance Native Drivers: SQL Links** with SQL 92 compliant access to Oracle, MS SQL Server, Sybase System10, Sybase System4, Informix, DB2, AS400, and InterBase. Native drivers provide the fastest way to connect to database servers by writing directly to Database Client Services.
- **Paradox and dBASE drivers**. These core database drivers can be queried using the new common database query engine. It also supports a full transactional model which allows multiple record updates to be grouped then rolled back or committed to the underlying table.
- **An ODBC socket** that allows access to any remote or PC data source for which an ODBC driver is available; such as Access, Terradata and /or, IMS datasources. Applications access all the features of Borland Database Engine even when using an ODBC driver. These features include: navigational access to data, new Cached Update transactional model, bi-directional cursors, and cross database operations.
- **Shared System Services** that provide mechanisms for data buffering, memory management; sort, query and batch functionalities; cursor and operating system abstractions; and more. These system services relieve the developer of many burdens, for example:

  1. Translate functions for seamlessly moving and scaling data from say, InterBase on NT to Oracle on a UNIX machine without having to translate data types. It also supports upsizing of data from PC tables in Paradox and dBASE.
  2. Memory or disk size allocations when caching database server result sets in order to prevent operating system errors.

  - **Borland Database Engine API**: Delphi Client / Server Suite 2.0 has encapsulated much of Borland Database Engine into high level objects that can be edited visually in the design environment. More advanced functionality can accessed via the complete API of the Borland Database Engine. This means that there are no barriers to writing extremely powerful database applications.
  - **Internationalization**: Applications are increasingly being created for an international market place to expand business potential and meet the needs of customers more readily. The Borland Database Engine has been designed to assist developers in creating and deploying applications for international markets. The Borland Database Engine comes with more than 50 language drivers. This helps an application developer in isolating language-specific help messages in a separate DLL. The sorted order, capitalization, accent and other language-specific features are supported by the Borland Database Engine as well.

B High Performance Native Drivers

SQL Link native drivers provide the fastest way to connect to database servers by writing directly to Database Client Services. Delphi Client / Server Suite 2.0 incorporates new 32 bit high speed native drivers that offer even faster performance and more functionality. The new SQL Links enables tools such as the SQL Monitor and SQL Explorer to assist the developer in creating and tuning applications. Highlights include:

- **Server Specific Optimizations:** affect the speed that applications open and affect the interaction between the end-user and the application.

- **New Functionality:** Delphi Client / Server Suite 2.0's new integrated SQL Explorer takes advantage of new SQL Link schema inquiry functions that give additional information about the database server: Integrity constraints, function lists, stored procedure definitions, domains, synonyms, and view definitions. The developer can now use Delphi Client / Server Suite 2.0 to administer all aspects of a complete Client / Server application.

- **Optimized Batch Operations:** The new SQL Links optimizes network traffic on batch operations by performing them on the server when source and destination tables are on the same machine. There is no need to move the data from the Server to the Local machine and back to the server. It is also possible for the developer to control how many modified records to include in a batch.

- **Governors** have been added to limit the number of rows that are transferred to the client by using the MAX ROWS SQL setting. This allows developers to work in a draft mode to speed the application development process.

C Transaction Models

Whether you have a large number of transactions, typically characterized as OLTP (On-line Transaction Processing) or a small number of transactions on your database server, Delphi Client / Server Suite 2.0, has the flexibility to meet these needs. The developer now has a choice of using a new Cached Update for more control and less server locking or the existing Navigational Update transactional model for tighter Client / Server coordination.

While updating a database server, the developer must be aware of locking mechanisms imposed by the server and the implications these locks have on users of an application. Locking is important because it creates contentions between users for ownership of a data record when doing everyday database operations: delete, modify, insert, etc. The longer a lock exists, the higher probability that a contention will occur and disrupt normal work-flow.

Another area for the developer to be aware of is the coordination between the server data and the client result set. The highest level of coordination means that the server will have the most up-to-date information at any one time whereby a lower level of coordination means that data on the server becomes out-of-date. The level of synchronization or detachment between the client and the server is now provided for in two ways with Cached Update or Navigational Update modes.

1 Navigational Updates

Navigational Updates are a mechanism by which each local transaction is sent individually to a database server within the context of a SQL Transaction. Navigational Updates complement Cached Updates and have two key benefits.

a Benefits

1. Navigational Updates ensure that the result set of a SQL query on a client is kept in tight coordination with the underlying tables on a database server by making transactions atomic. It maintains the highest level of integrity between the server and the client.

2. Navigational Updates with Optimistic Locking reduces the amount of time a record, table, index page or record page is in a lock state by maintaining a copy of the record on the client and updating a user request in a SQL transaction with a single record change (singleton transaction).

b *Implementation*

Delphi and the Borland Database Engine implement a locking strategy called *optimistic* locking. The client requests a remote record, and gets a buffered copy of it to work with without instantiating a lock on the server. When a user attempts to modify the record with a delete, insert or modify statement, a singleton transaction is initiated. Within this SQL Transaction, the copied result set record is compared to the data that's in the underlying table. This means that there are two possible results for the comparison and thus the singleton transaction.

1. If the comparison doesn't find any changes or discrepancies in the underlying record, your edits are posted to the table.

2. If the comparison finds a change to the underlying record in the table your changes are ignored and an exception is raised indicating that someone else modified or deleted the record while you were working on it. Your application can then implement custom exception handling code, for example, to log the errors in a database or retry the update.

Delphi offers options for comparing the client copy of the record and the underlying table data. The developer can do a check against the complete old row, check the unique ID plus changed columns, or check just the unique ID.

2 New Cached Updates

The Cached Update builds on the Navigational Update model. Cached Update batches a set of local transactions and sends them all at once to the server. This offers a degree of flexibility that matches your application design goals.

a *Benefits:*

1. When there are a high volume of users with many transactions being made to a database server, Cached Update significantly lowers the number of contentions that will occur thereby reducing the wait times for users.

2. Cached Update significantly lowers the number of record contentions imposed by Page Level Locking, Table Locking and Page Index Locking by reducing the amount of time that a contiguous or non-contiguous set of records are in a locked state.

3. Cached Update reduces the amount of network traffic between a client and a server by batching and packaging multiple communications into one communication.

4. Cached Updates enable the developer to separate user interaction from the underlying database records. This offers the ability to

intercept transactions before they reach the database server and substitute customized updating code. In this way, transactions can be further optimized and developers given more control. For example, by turning on cached updates, user interaction with a join query, an otherwise non-live query, can be intercepted and parsed into separate table transactions with query plans.

b *Implementation*

The Cached Update mechanism reduces record contentions by buffering server requests on the client; thereby not imposing any locks on the record, record page, index page, or table. The cache intercepts and stores inserts, modifies and deletes made by the user to any given record. The cached record updates are then batched to the server within the context of a SQL Transaction.

Locks are typically implemented on a database server to forbid the same record being read and modified at the same by two different users. Since no record locks are held before the Cached Update send operation, there is a possibility that the same records are also being updated by another user within the time scope of your off-line transaction. This means there are three possibilities for the successful completion of a cached update:

1. All cached operations are submitted successfully.

2. Any one cached operation fails and the rest of the cached operations are aborted and rolled back. An exception is then raised so that custom code can then be implemented.

3. Any one cached operation fails and the developer can choose to skip or fix it within a callback event.

This implementation means that the developer is in complete control.

D *Transactions and Delphi Exception Handling*

Responding to the result of a SQL Transactions is commonplace activity for Client / Server applications. Developers commit successful transactions and respond appropriately to unsuccessful transactions. Delphi marries object oriented language constructs with the database architecture to provide simple and robust transaction control via object oriented exception handling.

Robust Exception Handling: To make your applications robust, your code needs to recognize SQL exceptions, user exceptions and other system exceptions when they occur and respond to them accordingly. Delphi provides you with a mechanism to handle errors in a consistent manner that allows the application to recover without losing or corrupting data. Delphi pre-defines over 50 exception types (e.g. Access Violation, Math Error, Database errors) and allows the developer to define their own exception objects using inheritance of the base Exception class.

As a result, Delphi applications can recover from errors that in other p-code systems would result in an Access Violation; the result of which could lead to database corruption and frustrated users.

The combination of exception handling and transaction management provides for clean and readable code that is easily maintained. This code example below demonstrates the protection of a SQL transaction from any type of error without unnecessarily complex code to check the database server return codes, system resources, user errors, etc.

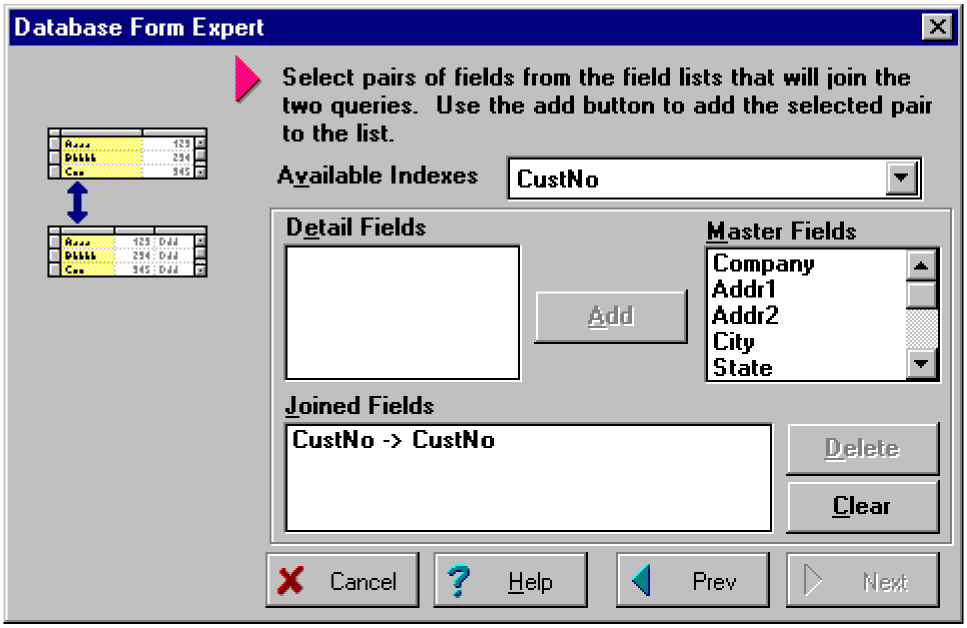
```
Procedure Transact_Except (Sender : TObject);
   begin
     try            { Should something fail in any code block statement, jump
                    immediately to the Except clause. }

        Database1.StartTransaction;
        Query1.SQL.Clear;
                                                   {subtract from selected account}
       Query1.SQL.Add
                    (Format ('update accounts' + 'set balance = balance - %s)
                    where acct_num = %s ', [edit1.text, Table1[acct_num]));
        Query1.ExecSQL;
        Query1.SQL.Clear;
                                                   {add to selected account}
        Query1.SQL.Add
                    (Format ('update accounts' + 'set balance = balance + %s)
                    where acct_num = %s ', [edit1.text, Table1[acct_num]));
        Query1.ExecSQL;
        Database1.Commit;

     except                                        {if error jump here}
         On e: DatabaseError do begin              {if database error do this}
                    database1.Rollback;
                    ShowMessage (e.message + 'Record changed before transaction')
        end;
        On MathError do begin                      {if math error do this}
                    database1.Rollback;
                    ShowMessage(e.message + 'Illegal Math Operation');
                    raise;
        end;
     end {except block}
end; {procedure}
```

Delphi makes it easy to create robust applications

E Heterogeneous Database Access

Presenting data as it pertains to the business need and not according to the location or type of data allows the corporation to create more flexible applications. The Borland Database Engine can present data to the developer as an abstract entity that is independent of machine type, database server, or data type. Developers can use this abstraction to present data from multiple database sources to the user independently of where it actually exists. It is possible to view information from ORACLE, Sybase, InterBase, Informix, Access, Paradox and Visual dBASE at the same time. It is also possible to present master-detail relationship between different database servers.

Delphi's Database Form Expert assists the developer in creating homogeneous or heterogeneous Client / Server database applications without writing complex code. It uses the Borland Database Engines linked cursor mechanism to create powerful master-detail database applications automatically generating the proper queries, forms and Delphi code.



Database Form Expert

Select pairs of fields from the field lists that will join the two queries. Use the add button to add the selected pair to the list.

Available Indexes: CustNo

Detail Fields

Master Fields

Company
Addr1
Addr2
City
State

Joined Fields

CustNo -> CustNo

Delphi's Database Form Expert creates master-detail applications easily

F *32-bit Windows Support*

Support for the native 32-bit operating systems means that you can take advantage of advanced features such as long file names, multi-tasking and resource allocations. The Borland Database Engine supports 32-bit Windows 95 and Windows NT platforms, including the following features:

- Multithreading within a single application. You can run multiple queries in the background within Borland Database Engine while using Borland Database Engine features in the foreground.
- Preemptive multitasking—multiple applications can run simultaneously and can access the same database files without work-flow interruption.
- Shared memory manager and shared buffer manager for lower resource usage and higher performance.
- Long filenames: You can give Borland Database Engine files long, descriptive names—up to 260 characters—that may contain spaces.

Multitasking queries and updates means you won't have to wait for naturally asynchronous operations. Taking advantage of the operating system means that applications can take on larger scope with more advanced functionality.

VI Conclusion

Delphi Client / Server Suite 2.0 is an application development tool designed to meet the Client / Server needs of corporations. Whether you are building an Executive Information System or a simple Business Automation System, Delphi Client / Server Suite 2.0's object oriented architecture delivers the highest level of flexibility, power and performance.

The Borland Database Engine offers the strongest foundation for building client /server database applications. The Delphi Client / Server Suite 2.0 component architecture includes high level objects that encapsulate the Borland Database Engine and allows the developer to concentrate on application building. The new Database Application Architecture takes these abstractions a step further and provides a solid framework for delivering reusable, scaleable and high performing applications in a team environment.

Delphi Client / Server Suite 2.0 meets the needs of the corporation, the end user and the developer by incorporating leading edge technology that let's you develop better applications in less time than with any other tools.