

Borland®
Delphi® 2.0

RADical Performance

Client/Server and Windows Development

Reviewer's Guide

March 1996

Copyright © 1996 Borland International, Inc. All rights reserved.

All Borland product names are trademarks of Borland International, Inc.

Other brand and product names may be trade marks or registered trade marks of their respective holders.

Feb 1996

PRINTED IN THE USA.

10 9 8 7 6 5 4 3 2 1

Table of Contents

Introduction

- Delphi 2.0 Product Line
- Delphi 2.0, the Client/Server Business Solution
- Delphi, The Solution for Developers
 - Delphi Development Today
- Delphi receives wide-scale acceptance
- Premier Value Added Partners Program
- Using the Delphi 2.0 Reviewer's Guide

Product Information

- Delphi Desktop 2.0
- Delphi Developer 2.0
- Delphi Client/Server Suite 2.0

Delphi Features and Benefits

- Performance
- Productivity / Ease of Use
- Advanced Database Capabilities
- Windows 95 and NT System Support
- Client/Server Application Development

Quick Tour of Delphi 2.0

- Rapid Application Development
 - Windows 95 User Interface
- Database Speed and Scalability
 - Delphi 2.0's new Data Module Objects
 - Object Repository
 - Adding a Lookup
 - Business Rules
 - Compiler
 - Visual Form Inheritance
 - SQL Database Explorer
 - Enhanced DataGrid with Codeless Lookups
- Filters
 - DataListBox and DataComboBox
 - Multi-Object Grid

Reporting with Delphi 2.0

ReportSmith
QuickReports

Part Two: Technical Details

Compiler Optimizations

- New Optimizing Linker

- Increased 32-Bit Capacity

- New 32-Bit Data Types

- Preliminary Benchmark results

- The Advantage of Native Code Compilers

Application Management and Debugging

- Improved Compiler Error Messages and Diagnostics

Reuse and Maintainability

Object Pascal

Visual Component Library - Building New Components

Client/Server Solutions: Delphi's Database Architecture

Borland Database Engine

- Logical Application Distribution:

 - Network OLE and VB4's Remote Automation technology

Scaleable Database Dictionary

Connectivity and Transactions

- Cached Updates

- Update Mode

SQL Monitor

Exception handling: The path to robust applications

- Implementing Exception Handling

- try..finally

InterBase NT - Relational Database

Local InterBase Server

OLE Automation and OLE Controls (OCXs)

- The Importance of being OLE

- Office Integration

- Creating OLE Automation Servers

- Using OLE Controls (OCXs)

Conclusion

Appendices

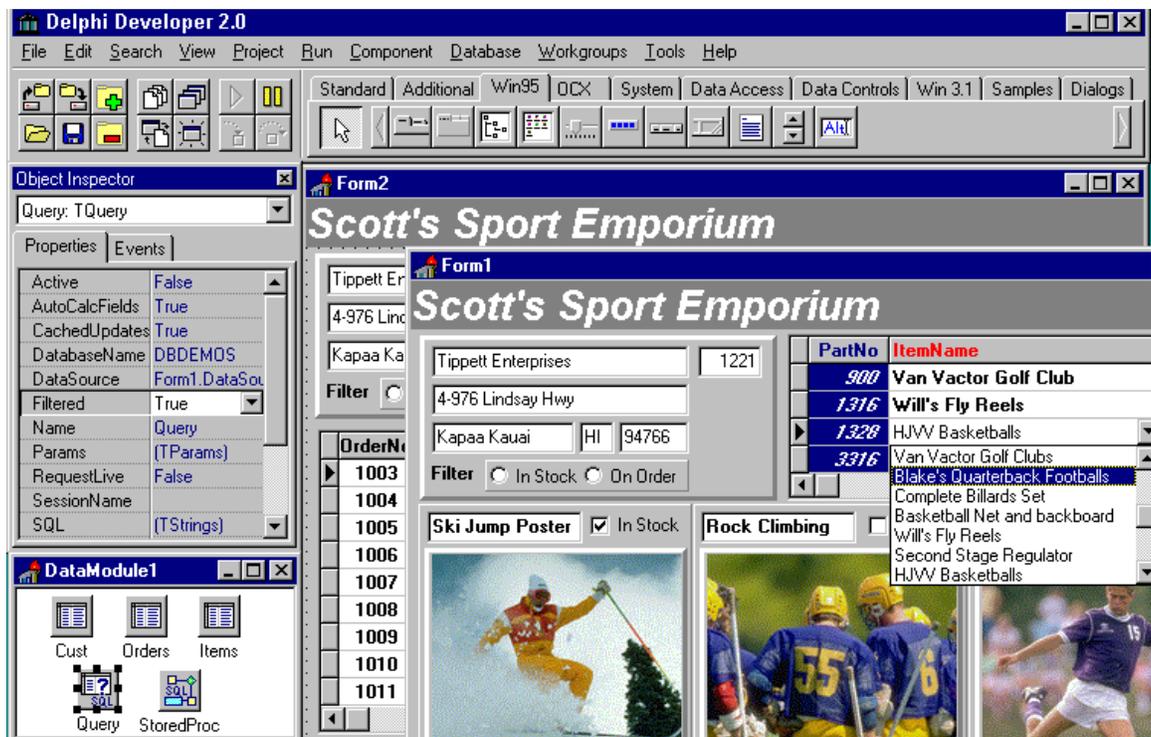
Documentation Overview

Hardware/software requirements

Introduction

Welcome to the Product Reviewer's Guide for the next generation of Borland's high-performance Windows development tool, Delphi 2.0. Since the introduction in February 1995, Borland's Delphi and Delphi Client/Server development tools have set a new standard in high-performance rapid application development. As a result of Delphi's unique combination of a native code compiler, visual two-way tools and scalable database technology, Delphi has won over 2 dozen awards worldwide and has become the fastest growing client/server development tool.

Delphi 2.0 is a strategic tool designed to give corporate and government organizations greater returns on their IT investments and significant competitive advantages. Delphi 2.0 builds on Delphi 1.0's strengths to continue as the highest performing RAD tool while reducing development time for robust, flexible applications.



Delphi 2.0 uses an object-oriented component architecture to create robust, reusable code.

Delphi 2.0 Product Line

This guide provides an overview of the Delphi product line, its major features and benefits, and a look at the underlying technology that makes Delphi unique. The Delphi 2.0 product line includes:

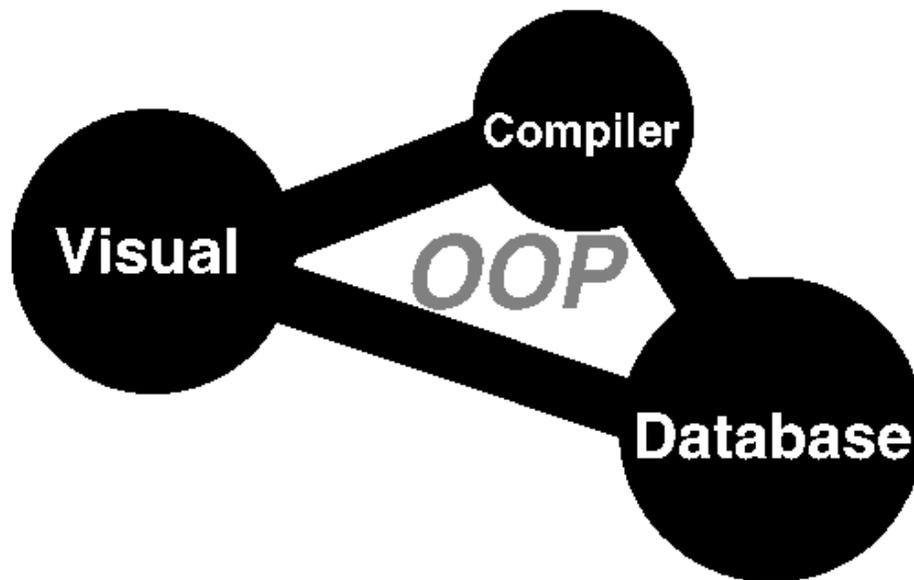
- **Delphi Client/Server Suite 2.0** the high performance, Windows 95 and NT, rapid application development system with a full suite of client and server tools to build complete client/server solutions.
- **Delphi Developer 2.0** targeted to professional developers who want to develop high-performance desktop and LAN-based applications. Includes local /

LAN database support and full scalability to Delphi Client/Server Suite 2.0.

- **Delphi Desktop 2.0** the fastest, easiest way to create Windows 95 and NT applications. Includes local database support, and a complete set of visual components for rapid application development.

Delphi 2.0, the Client/Server Business Solution

By combining the ease of use of a visual development environment with Borland's high performance compiler and database technology, Delphi offers optimal performance, scalability, and rapid development. Delphi's unique architecture provides breakthrough technology for today, and a powerful foundation for future evolution. It's open architecture enables Delphi to accommodate new tools, new components, new operating system facilities and even higher levels of performance and productivity through application partitioning.



Delphi 2.0 combines a high performance compiler, visual development, and database scalability in a robust object-oriented environment.

Delphi 2.0 continues to be the performance leader while it builds on Delphi 1.0's successful object oriented architecture. The needs of application developers map directly the design goals for Delphi 2.0:

- **Insure high performance applications** by further increasing Delphi's performance advantage with a new 32-bit optimizing compiler.
- **Shorten development cycles** by offering maximum reusability and maintainability through object oriented programming and rapid application development tools.
- **Enhance database speed and scalability** with high performance 32 bit database drivers, and a robust, scaleable database architecture.

- **Provide flexible operating system support** with full support for 32 bit Windows 95, Windows NT, and 16 bit Windows 3.1.
- Provide **full compatibility** for existing Delphi applications.
- Leverage Microsoft system technologies (OLE Automation, OCX support and ODBC support) for **easy integration** with other Windows applications, including Microsoft Word and Excel, Paradox, Quattro Pro and others.
- Offer a **suite of Client/Server tools** that provides a complete solution for client and server support.
- Facilitate **code maintainability** by advancing Delphi's **object oriented** development environment with Visual Form Inheritance.

Delphi, The Solution for Developers

Since change is inevitable, software developers need speed and flexibility to create applications for a wide array of client needs. Delphi offers the fastest way to the fastest applications. Delphi 2.0 supports application development for:

- **Windows 95, Windows NT and Windows 3.1.**
- **Client/Server** environments including Oracle, InterBase, Sybase, MS SQL, and data available through ODBC.
- Local and LAN based database applications.
- Building stand-alone **EXEs, DLLs, or OLE Automation controllers and servers** for use in other Windows applications.
- Graphical, visual and **multimedia** applications. A number of Third Party products are available for Delphi development on the World Wide Web.

Delphi Development Today

The Banking, Financial, Chemical, Consulting, Communication, and Multimedia industries are using Delphi for a broad range of applications that demand rapid development and high performance. Delphi customers include such organizations as

- | | |
|-----------------------|----------------------------------|
| • Alcatel | • EDS |
| • American Stores | • Ernst & Young |
| • Arthur Anderson | • Fiat |
| • AT&T | • First National Bank of Chicago |
| • BP Shipping | • Glaxo |
| • Bank of America | • KPMG |
| • BBC Television | • Mercury Communications |
| • British Telecom | • Netscape |
| • City of Los Angeles | • Sarah Lee Knitting |
| • Compaq | • Standard & Poors |

- Conoco
- Coopers & Lybrand
- DHL
- Dover Elevators
- SwissBank SG Warburg
- Union Bank
- US Marine Corps
- Siemens and many others

Delphi has been used to build applications in a variety of industries. For example,

- 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 a ***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 publisher of technical books and magazines has used Delphi to create a full motion ***multi-media application*** to break into software publishing.
- A national health agency use Delphi to create a ***Dynamic Link Library (DLL)*** to extend an existing Paradox for Windows application in order to perform real-time data collection through scientific instruments for measuring Radon gas levels.
- A Fortune 100 organization is using a Delphi Client/Server application and Microsoft SQL as part of a major ***business re-engineering*** effort to dramatically increase efficiency and increase customer satisfaction.
- 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 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.



Several case studies are available separately that describe these and other applications in more detail.

Delphi receives wide-scale acceptance

Delphi has achieved a tremendous level of third party support including dozens of add-on libraries and compatible tools, more than 40 books, a half dozen monthly magazines and newsletters, and many training courses supported by a growing number of third party consultants.



The *Delphi Power Tools* Catalog contains information on hundreds of services including training centers, user groups, products and books.

Premier Value Added Partners Program

Borland's new Premier Value Added Partners Program provides major corporations and government agencies with applications, consulting and training services for Delphi Client/Server Suite 2.0. Partners supporting Delphi include:

- Arthur Anderson Business Consulting
- Adams & Reynolds
- ANATEC
- Client/Server
- Computer Generated Solutions
- Compuware
- Decision Consultants Inc.
- Dann Systems
- Erudit Software
- Financial Dynamics
- Grant Thornton LLP
- Integrated Systems Consulting Group
- Interlink Group, Inc.
- LINC Systems Corporation
- Mustech Corporation
- Raymond James Consulting

Using the Delphi 2.0 Reviewer's Guide

This guide is divided into two parts. Part One contains a complete explanation of the new features of Delphi 2.0 and details the product line. Part Two contains in depth technical discussions of Delphi's more advanced concepts like building components and creating OLE Automation Servers and Controllers.

Part One contains:

- **Product information** about the three new versions of Delphi 2.0.
- **Feature and Benefit** descriptions in each of Delphi's major focus areas: performance, reusability, database scalability, rapid application development and object orientation.
- A **Quick Tour** of the Delphi environment that describes how many of the new features are used in application development.

Part Two details some of the more advanced features of Delphi 2.0 including:

- Compiler Optimizations
- Application Management and Debugging
- Exception handling: The path to robust applications
- Visual Component Library - Building New Components
- Client/Server Solutions: Delphi's Database Architecture
- InterBase NT - Relational Database
- OLE Automation and OLE Controls (OCXs)

For further research, the following white papers are available:

- *An Overview of the Delphi 2.0 Optimizing Native Code Compiler for Windows 95 and NT*
- *An Overview of the Delphi 2.0 Windows 95 Common Controls*
- *The Delphi Open Tools Interface*
- *Delphi 2.0 Client/Server Database Architecture*
- *Delphi Client/Server Suite 2.0 Integrated Version Control Management*
- *Delphi and Corporate Return on Investment*
- *Delphi 2.0 Benchmark Results*

Product Information



Delphi Desktop 2.0

Delphi Desktop 2.0 is the easiest way to create the fastest applications for Windows 95 and Windows NT. It combines the most intuitive, object-oriented development environment with over 90 customizable, reusable components for immediate productivity. Drag and drop database tools provide an innovative interface for building 32-bit applications.

Delphi Desktop 2.0 applications run up to 300-400% faster than 16-bit Delphi, and up to 15-50 times faster than those built with p-code interpreters. Delphi 2.0 leverages Windows' 32-bit architecture, adds an Object Repository, and supports the reuse of data modules – once again raising the bar for application development tools. All Delphi 2.0 version include the 16-bit Delphi 1.0 for Windows 3.1.

<i>Delphi Desktop 2.0 for the individual Windows 95 and NT Developer</i>	Delphi Desktop	Delphi Developer	C/S Suite 2.0
High performance, 32-bit optimizing native code compiler	Y	Y	Y
Over 90 components for rapid application development	Y	Y	Y
Complete component suite of Windows 95 common controls	Y	Y	Y
Data Aware components to build local database applications	Y	Y	Y
Data Modules for centralized data integrity and business rules	Y	Y	Y
Object Oriented, fully extensible component architecture	Y	Y	Y
Visual Form Inheritance and Form linking	Y	Y	Y
Full Windows 95 support for OLE automation, and OCXs	Y	Y	Y
True 32-bit development with full access to the Win32 API	Y	Y	Y
Database Explorer to modify tables, aliases and indices	Y	Y	Y
Object Repository to store and reuse objects and forms	Y	Y	Y
Creates high-performance reusable DLLs and EXEs	Y	Y	Y
Integrated Development Environment	Y	Y	Y
QuickReports for fast, integrated Delphi reports	Y	Y	Y
Includes the full 16-bit Delphi 1.0	Y	Y	Y
Complete manual set including <i>Getting Started, Delphi User's Guide, Component Writer's Guide, Object Pascal Language Guide, Database Application Developer's Guide</i>	Y	Y	Y



Delphi Developer 2.0

Delphi Developer is the fastest way to build 32-bit professional multi-user applications for Windows 95 and NT. It is the next step for Delphi 1.0 owners. Since Delphi 1.0 and Delphi Developer 2.0 are code compatible, no matter where your Windows based applications reside today, Delphi provides a complete solution.

In addition to the features found in Delphi Desktop, Delphi Developer includes: a scaleable Data Dictionary, Multi-Object Grid, complete ODBC support, source code to over 100 native Delphi components, sample OCXs, an expanded Open Tools API, ReportSmith, Local InterBase Server and much more. Includes 16-bit Delphi 1.0 for free.

Delphi Developer is targeted to professional developers who want to develop high-performance desktop applications accessing local and LAN databases, including: dBASE, Paradox, Local InterBase and ODBC.

<i>Delphi Developer 2.0 includes these additional features for the professional developer</i>	Delphi Desktop	Delphi Developer	C/S Suite 2.0
Everything included in Delphi Desktop 2.0	Y	Y	Y
Scaleable Data Dictionary		Y	Y
Multi Object Grid		Y	Y
Over 100 VCL components for rapid application development		Y	Y
32-bit ReportSmith, for high volume client/server reporting		Y	Y
BDE low-level API support and Help Files		Y	Y
ODBC Support		Y	Y
Single User Local InterBase Server		Y	Y
InstallShield Express for easy installation and deployment		Y	Y
Winsight32 for monitoring windows messaging		Y	Y
Expanded Open Tools API		Y	Y
Team Development Interface (Requires Intersolv PVCS)		Y	Y
Visual Component Library Source code and complete manual		Y	Y
Full documentation of Delphi Desktop plus <i>ReportSmith Creating Reports, Delphi Reference Library Guide, InterBase Server Getting Started</i>		Y	Y



Delphi Client/Server Suite 2.0

Delphi Client/Server Suite 2.0 contains everything you need to build and deliver high-performance client/server applications. Unmatched performance, data integrity, and code reuse are all contained in a robust object-oriented interface, that maximizes productivity across the enterprise. With a host of tools for optimized client/server development, Delphi Client/Server Suite 2.0 offers a complete solution.

The suite also includes a 2-developer copy of InterBase for Windows NT, a fast and efficient SQL database server; a complete set of database design and analysis tools; integrated team-development support; and native 32-bit SQL Links for royalty free deployment on Sybase, Oracle, InterBase, and SQL Server.

Delphi Client/Server Suite is targeted to corporate developers, departmental programmers, VARs, system integrators, consultants, and ISVs who want to develop high performance workgroup and client/server applications.

<i>Delphi Client/Server Suite 2.0 includes these additional features for professional client/server developers</i>	Delphi Desktop	Delphi Developer	C/S Suite 2.0
Everything in Delphi Desktop and Delphi Developer		Y	Y
High performance 32-bit SQL Links native drivers for unlimited deployment			Y
SQL Database Explorer to browse server meta data			Y
SQL Monitor for testing, debugging and performance tuning			Y
2 user InterBase NT License			Y
Data Pump Expert for rapid upsizing and application scaling			Y
Integrated Intersolv PVCS Version Control			Y
ReportSmith SQL edition			Y
Visual Query Builder to easily create SQL queries			Y
Cached Updates			Y
Client/Server Documentation			Y
Includes 16-bit Delphi Client/Server 1.0 for Windows 3.1			Y
Full documentation of Delphi Developer plus <i>Getting Started SQL Links, InterBase Language Reference, InterBase Data Definition Guide</i>			Y

Delphi Features and Benefits

Delphi 2.0 encompasses a large variety of new features covering performance, reusability, rapid application development and database scalability. The following section presents the features *new* to Delphi 2.0.

Performance

Delphi's 32-bit high performance compiler and language enhancements include:

- ✓ The world's fastest high-performance 32-bit optimizing native code compiler
- ✓ Full code compatibility
- ✓ True 32-bit development
- ✓ Easier code sharing with other Windows applications
- ✓ Object oriented component architecture
- ✓ Visual Two-way tools
- ✓ Graphical debugger
- ✓ Structured, object oriented language

High-performance 32-bit optimizing native code compiler

Delphi's optimizing 32-bit native code compiler gives you the fastest way to the fastest applications. Your applications result in royalty free, standalone executable files for rapid deployment. No runtime interpreter DLLS are necessary. For Delphi users, a simple recompile can increase performance up to 300-400%. Compared to interpreted P-code, applications run up to 15-50 times faster. The new optimized linker makes EXEs up to 20-25% smaller. Other new 32-bit optimizations include: register optimizations, call stack overhead elimination, loop induction variables, Pentium safe FDIV, and common sub-expression removal.

Full code compatibility

Our goal with Delphi 2.0 is to offer full compatibility with 16-bit code. In order to migrate code from the 16-bit version to the 32-bit version, it must be recompiled, but otherwise, few changes are necessary. In most cases, developers can simply load their 16 bit applications into the new environment, compile the code and immediately see up to a 300 - 400% performance boost.

In some cases, conversion of code may be necessary, but only where fundamental assumptions are made that depend on representations that have changed in either the Windows environment or code that is dependent upon the low-level physical implementation of 16-bit data types. Delphi automatically handles changes in the Windows message types.

No matter where your Windows-based applications reside today, Delphi supplies the complete solution: stay with Windows 3.1, or migrate to a high powered 32-bit development environment on Windows 95 or NT. Delphi's code compatibility from 16 bit up to 32-bit lets you protect your application investments while moving to new operating systems. To ease migration, each version of Delphi 2.0 includes the current 16-bit version of Delphi 1.0 for Windows 3.1.

True 32-bit Development

Because Delphi 2.0 is a native code 32-bit compiler, you gain powerful performance benefits, faster applications and support for all new platform features including long filenames, Unicode support, multi-threading, and complete access to the Win32 API. There's more available memory because the compiler runs in a 32-bit flat address space, completely eliminating the limitations associated with a 16-bit segmented architecture like Windows 3.1. You can declare arrays, strings, records and other data structures to be as large as you like, up to 2 gigabytes in size! The size is virtually limitless because there are no more 64K barriers.

Easier code sharing with other Windows applications

Because of Delphi's native code compiler, you can create reusable high performance DLLs that work with C++, dBASE, Paradox, Visual Basic, PowerBuilder and other Windows applications. Delphi's component based architecture lets you seamlessly integrate DLLs, OCX controls and OLE Automation servers into your Windows programs. Delphi 2.0 is built around the same high performance back-end compiler technology as Borland C++, with the benefit of allowing customers to generate .OBJ files or COM (Common Object Module) objects and maintain closer integration of C++ and Delphi applications. Programmers can leverage existing work because the common backend lets them share code between Delphi and C or C++.

Object-Oriented Component Architecture

Delphi lets you exploit the power of object-oriented programming to create robust, efficient applications. Build your own components with Delphi's proven object-oriented component architecture. Because Delphi 2.0 is a completely object-oriented environment, integration of OLE controls (OCX) is seamless. Use, customize or subclass components to fit your development needs – all from within the Delphi environment.

Visual Two-way tools

Because Delphi is a two-way tool, it means that developers can always see the underlying code as they develop visually. As a result, developers don't have to worry about "hitting the wall," running into limitations with Delphi since they can always "get down to the code". Because of Delphi's two-way tools, it seamlessly supports team development through version control and other third party tools. Delphi 2.0 introduces a single keystroke to toggle between the visual and code representation of a form.

Structured, Object-Oriented language

Delphi uses a structured object-oriented language, Object Pascal, that provides the ease of programming of a high-level 4GL and the performance and power of a 3GL. Programmers can become immediately productive since Delphi eliminates the need to learn complexities of the underlying Windows event model. Delphi fully supports advanced programming concepts including encapsulation, inheritance, polymorphism and exception handling. Because Delphi uses a native code compiler, you can create efficient standalone .EXEs as well as reusable Dynamic Link Libraries (DLLs) that can be called from other applications or tools.

Graphical Debugger

Delphi has a powerful Graphical Debugger that makes it easy to find and fix bugs in your code. You can set breakpoints, examine and change variables, view thread status, single step through the code, and understand its exact behavior. For advanced debugging requirements, you can use the separately available Turbo Debugger to view the underlying assembly language code and machine registers.

Productivity / Ease of Use

Enhancements to rapid application development, object reusability and code maintainability include:

- ✓ Visual Component Library (VCL32)
- ✓ Easier to understand error messages
- ✓ Visual Form Inheritance
- ✓ Multi-error compile with Hints and Warnings
- ✓ Object Repository
- ✓ Installation and Deployment expert*
- ✓ Extended Open Tools API*
- ✓ Advanced math library*
- ✓ VCL32 Source Code*

** Available only in Delphi Developer 2.0 and Delphi Client/Server Suite 2.0.*

Visual Component Library (VCL32):

Delphi 2.0 includes an enhanced implementation of the Visual Component Library, containing over 100 reusable components that developers “drag and drop” to create sophisticated Windows applications. VCL32 includes full encapsulation of the Windows 95 user interface elements (Tree Views, Trackbars, sliders, progress bars, Toolbars, Rich Edit, List Views, Image Lists header and status bar controls, etc), along with many additional components. These additional components complement the existing user interface, data aware and multimedia tools to help you build sophisticated applications and move from prototype to production quickly.

Visual Form Inheritance

Visual Form Inheritance gives you the power of object oriented programming in an easy to use visual tool. Visual Form Inheritance centralizes changes, makes code easier to maintain, and reduces turn-around time for new features. A key strength of Delphi is its proven architecture for object oriented computing and reusability; using inherited forms extends this model to create flexible applications that rapidly adjust to changes in production environments.

Easier to understand Error messages

The compiler now gives a much clearer, English language indication of the problems with better diagnostic messages on syntactic errors. Simpler error messages make Delphi easier for programmers who are new to Object Pascal. The easier to understand error messages include many common syntactic errors like forgotten semi-colons. These messages provide a smooth transition for programmers using Delphi to complement their existing development language.

Multi-error compile with hints and warnings

The new multi-error compiler architecture helps programmers resolve errors faster. Using a hot-key, programmers can jump to and fix multiple problems in a single compile. The new compiler even catches logic errors, like empty loops, uninitialized pointers and unused variables that result from ambiguous or incorrect code. This makes it easier to find bugs before they cause problems, and to verify large programs for correctness.

Object Repository

Delphi 2.0 introduces a next generation methodology for storing and reusing data models, business rules, objects and forms. A centralized object repository of form designs links into a reusable database architecture. The repository gives developers unprecedented responsiveness to application demands. Because building on an existing foundation reduces development time, Delphi's Object Repository stores forms and complete applications for reuse. Any new application can inherit, reference, or simply copy an existing structure – you pick the architecture that best fits your development needs.

Installation and deployment Expert *

Delphi supports fast and easy deployment of your completed application with an integrated expert that allows developers to create installation disks for their applications. InstallShield Express, Delphi's custom install utility, asks a few simple questions and then does the work of building a professional looking install for all applications - including efficient deployment of complex database applications.

Extended Open Tools API*

The Open Tools API allows for seamless integration with a variety of third party tools to support CASE, Object Oriented Analysis and Design, Testing, Transaction Process Monitors, etc. Delphi integrates with the latest leading-edge technologies to provide developers with an environment with unparalleled productivity. This also means that Delphi can integrate seamlessly with other development packages that you wish to use.

VCL32 Source Code*

In order for professional developers to more rapidly create their own reusable custom components, Delphi 2.0 Developer includes source code for the enhanced VCL32, which provides unique insights into the workings of Delphi and its components. Source code availability lets you learn from examples and reuse existing components.

Advanced math library*

Delphi 2.0 contains over 2 dozen functions for statistical and financial calculations, including source code with Pentium optimizations.

** Features available in Delphi Developer and Delphi Client/Server Suite only*

Advanced Database Capabilities

A Delphi database application is built using data-access components and data-aware user interface components. Delphi components communicate with the

Borland Database Engine, extract information from the Data Dictionary, and define business rules stored in Data Module Objects.

Delphi 2.0 includes an integrated suite of tools for building high-performance applications with such extensive features as:

- ✓ Data Module Objects
- ✓ Drag and drop database development
- ✓ Enhanced DBGrid
- ✓ 32-bit Borland Database Engine
- ✓ Data Aware Controls:
DbLookUpListBox,
DbLookUpComboBox
- ✓ Fast filters and smart lookup fields
- ✓ Database Explorer
- ✓ QuickReports
- ✓ ReportSmith*
- ✓ Multi Object Grid*
- ✓ Scaleable Database Dictionary *
- ✓ Local Interbase Server *

** Features available in Delphi Developer and Delphi Client/Server Suite only*

Drag and Drop Database Development

Delphi 2.0's makes powerful database programming as simple as drag and drop. Delphi uses the data dictionary to automatically customize the display and edit properties of your data. You select the fields and Delphi instantly builds all the connections for a live database application.

Data Module Objects

Data Module Objects act as your applications information core, they define data access and apply business rules to your application. With this centralized model for data access, you can separate business rules from GUI development for easier maintenance. When changes occur in your business model, simply change the data module, and results appear instantly throughout the organization. For faster database development, Delphi Data Modules act as a codeless way to connect and manage multiple forms from a single location. Working with data modules means you only have to define your database links once, and then drag and drop fields to build additional forms anywhere on your network.

Scaleable Database Dictionary

The Database Dictionary stores and uses customized information about the contents and behavior of the data in your tables. The data dictionary holds information about extended field attributes like minimum, maximum and default values, display preferences and edit masks. Because the dictionary is reusable across forms and applications, you can quickly establish and maintain data integrity.

Automatic Lookup Support

Delphi 2.0 adds new controls to its comprehensive suite of data aware components. Designed for fast and flexible programming, new data aware controls include: a variety of lookup and listbox capabilities to show your users exactly the information you want – while underlying tables remain free to update and display.

Enhanced Grid Control

The new custom grid control offers codeless support for quick searching and automatic lookup fields. A simple UI provides customization of display properties including fonts and colors.

32-bit Borland Database Engine

Delphi 2.0 is built on a new 32-bit Borland Database Engine. This flexible database engine scales smoothly to client/server environments while providing transparent data sharing between applications. With features like a new Query Engine, BCD support for financial applications and low-level API support, BDE forms a solid foundation for all your database applications. Borland's database engine is tightly linked into Delphi 2.0's reusable database repository and Data Module architecture.

Fast Filters and Smart Lookup Fields

Fast Filters and Smart Lookups let Delphi do the database work for you. Now you can find and display the information you need with the touch of a button. Fast Filters make viewing a specific subset of your data as fast and easy as changing a property. The Filter property supports flexible expression filters. Smart lookups automatically provide a dropdown box to seamlessly display data from multiple tables. Working through the Borland Database Engine, both fast filters and automatic lookups instantly present a customized view of your data.

QuickReport Components

QuickReports lets you design reports visually with Delphi components and the Delphi form designer. Reports are built with bands and may contain titles, page headers and footers, multiple detail sets, summaries, group headers and footers. Report from any DataSource, including Tables and Queries. Use the powerful on-screen preview to check your results. Summary and grouping bands perform summary and count calculations automatically.

Database Explorer

The Database Explorer provides a graphical way of managing all your database demands, it supports the creation and modification of tables, indices and aliases. It is an integrated database schema and content management utility tailored to the needs of professional database developers.

Local Interbase Server*

InterBase is a relational database management system that provides rapid transaction processing and data sharing in a single or multi-user environment. This single user version resides on the developer's PC and is commonly used as a local database environment to develop stand-alone desktop SQL applications and client/server applications with remote server connections. Any applications developed with the Local InterBase Server are easily scaled to the full multi-user server version running on NT, Windows 95, Novell Netware or Unix.

Multi Object Grid*

Showing detailed views of your data has never been easier. Delphi's new multi object grid lets you decide whether data appears in rows or individual panels. Panel views support multiple dropdowns, checkboxes, and memo fields. As with

the grid control, the multi-object grid supplies the user with codeless support for quick searching and automatic lookup fields.

ReportSmith*

Provides the most advanced report writing tool for developers who need to create sophisticated reports accessing large volumes of data. Includes the unique use of “live data” so that developers can see exactly how their report will look at design time.

** Features available in Delphi Developer and Delphi Client/Server Suite only*

Windows 95 and NT System Support

Delphi 2.0 easily shares information with other applications because of its support for:

- ✓ Standalone EXEs
- ✓ Reuseable DLLs
- ✓ Leverage the advantages of Windows 95 and NT
- ✓ Complete access to Windows 95 and NT API
- ✓ OLE Controls (OCXs)
- ✓ Sample OCX components
- ✓ OLE Automation Server *and* Controller support
- ✓ Full OLE Automation support for languages like Word or Excel Basic
- ✓ Multi-Threaded applications

Delphi 2.0 fully supports true 32-bit application development on both Windows 95 and Windows NT. Because application development environments are often a mix of NT, Windows 95 and Windows 3.1, Delphi 2.0 has been certified on both Windows 95 and NT and each version of Delphi 2.0 includes the full 16-bit Delphi 1.0 for development on Win 3.1.

One goal with Delphi 2.0 has been complete adherence to Microsoft system standards to ensure that developers can use Delphi 2.0 to create a diverse range of applications without limitations. Microsoft’s OLE technology includes a variety of important capabilities for increasing the ability of developers to create more modular and integrated applications. Delphi 2.0 goes beyond simply adhering to Microsoft standards; it actually makes the use of OLE technology easier through object orientation.

Leverage the advantages of Windows 95 and NT

Delphi 2.0 moves rapid application development into the 32-bit world. With a suite of new Windows 95 components, you can design elegant Windows 95 logo compliant applications. In addition to 32-bit support for long file names, multi-threading and the Windows 95 API, Delphi includes a complete suite of Windows 95 controls: TreeViews, Header controls, Status Bars, progress meters and more.

Complete access to Windows 95 and NT API

Delphi 2.0 allows programmers to exploit all of the enhancements of Windows 95 – including MAPI, Plug’n’Play, multi-threading, and Unicode – by allowing for direct calls to any Windows 95 or NT API.

OCX support

Delphi 2.0 fully supports the 32-bit OLE Custom Control (OCX) standard for add-in software components. Developers can add third-party or custom-made OCXs to the component palette quickly and easily, all OCX properties immediately appear in Delphi's object inspector. Because Delphi is object oriented, you can customize OCX's via inheritance.

In addition, because Delphi 2.0 is a native code compiler, advanced developers are able to write their own OLE controls (OCXs) within Delphi 2.0 though this is more difficult than creating Delphi components.

Sample OCX components*

Versatile graphing, charting, spreadsheet, and spell-checking OCXs are included for immediate use in applications. (Developer and C/S only)

OLE Automation Server and controller support

Use OLE automation to create or control other applications such as Microsoft Word and Excel, Lotus 1-2-3, dBASE, and Paradox. Because Delphi 2.0 can create high performance OLE automation controllers *and* OLE automation servers, Delphi 2.0 will allow developers to create partitioned applications easily with Network OLE when that technology becomes available. Delphi 2.0 fully supports in-process and out-of-process local OLE automation servers. Use Delphi 2.0 with the remote automation technology included in Visual Basic 4.0 and you get the added advantage of faster performance due to our optimizing native code compiler.

Multi-threading

Because Delphi 2.0 is a native code compiler it can take advantage of any platform feature on Windows 95 or Windows NT. This includes complete support for features such as multi-threading. You can easily create multi-threaded applications by selecting New Thread Object from the Delphi Object Repository. This will generate a unit with an object that inherits from the TThread type to simplify the creation of multi-threaded applications. You also have direct access to the threading API. For example, you can call the CreateThread API function with an Object Pascal function as a parameter. Similarly, you can set a thread priority by calling the SetThreadPriority API function.

The Delphi Visual Component Library (VCL) includes support for creating thread-safe applications. For example, the Synchronize method of the TThread class allows you to ensure that manipulation of VCL components is done safely within a thread without any possibility of conflicts in any other thread. Delphi 2.0 also introduces a new reserved word **ThreadVar** which allows you to declare thread-local storage, giving you complete control over variables used in different threads.

Client/Server Application Development

To adapt in rapidly changing business environments, developers must demand more from their client/server development tools. Delphi Client/Server Suite 2.0

offers the power and flexibility necessary to build reliable client/server solutions. Because a client/server solution is more than just a high performance development environment, Delphi 2.0 includes a comprehensive suite of tools for application maintenance, object sharing, and performance monitoring. Delphi Client/Server Suite 2.0 provides a complete client/server development solution. Delphi Client/Server Suite 2.0 includes:

- ✓ 32-bit SQL Links with unlimited deployment
- ✓ 2-Developer license InterBase NT
- ✓ Integrated Version Control
- ✓ Cached Updates
- ✓ Visual Query Builder
- ✓ SQL Explorer
- ✓ SQL Monitor

All features available only in Delphi/Client Server Suite 2.0

32-bit SQL Links with unlimited deployment

Delphi 2.0 enhances its' suite of high-performance 32-bit SQL Link native drivers for Oracle, InterBase, Sybase, and Microsoft SQL Server it introduces new 16 bit Sybase System 10 and DB2 Native drivers. Delphi Client/Server Suite includes an unlimited deployment license for all drivers.

InterBase NT Server 2 user license

InterBase is an SQL relational database management system that provides exceptional portability and scalability while maintaining high productivity and ease of use. InterBase offers excellent concurrency due to its Multi-generational Architecture, and an industry standard ANSI SQL 92 programming interface. Off-line application development can be done in Delphi's Local Interbase Server and easily ported to this full InterBase NT version.

SQL Explorer

The SQL Explorer allows developers to browse server-specific meta data including stored procedure definitions, triggers and index descriptions.

SQL Monitor

The SQL Monitor assists in testing, debugging and performance tuning SQL queries. It enables developers to trace calls between the client and server and report back the amount of time spent on a single operation. The traced information surfaces problematic SQL statements and helps to optimize transactions. When you need thorough testing, simply print or save the session log.

Cached Updates

Cached Updates significantly speed server responsiveness by reducing the amount of network traffic between a client and a server. Batched and packaged, multiple communications behave as a single transaction, thereby reducing server contention and improving application performance.

Visual Query Builder

Delphi 2.0 Client/Server Suite includes a 32-bit implementation of the powerful Visual Query Builder. This graphical SQL query technology automatically generates bug-free ANSI SQL-92 commands.

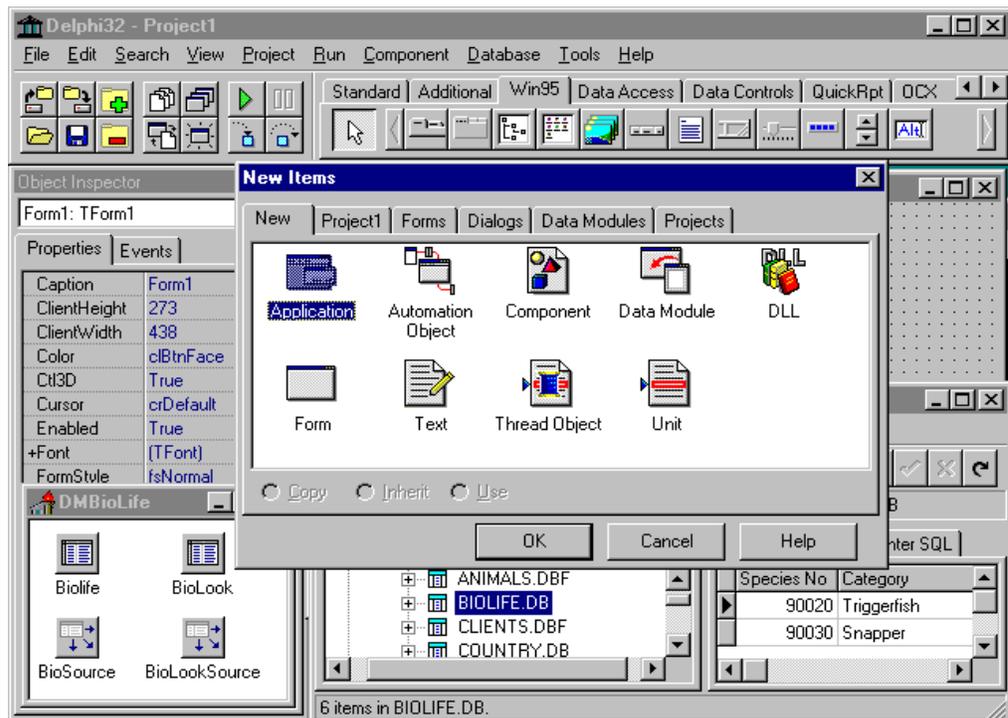
Integrated Version Control

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.

Quick Tour of Delphi 2.0

Designed for rapid application development, Delphi 2.0 helps you build applications to gain and maintain a competitive advantage by quickly moving your applications from prototype to production. To illustrate the power and flexibility of Delphi, let's take a quick tour and build an application. To develop with Delphi you need to understand its core development tools:

- **Component palette**, over 100 reusable components to build applications. Delphi Developer and Client/Server Suite include the full source code to the components in the 32-bit Visual Component Library.
- **Object Repository**, a central location for all reusable objects like forms and data modules. Sharing of application objects reduces development time.
- The **Object Inspector** lets you set the Properties of objects visually without writing code. It shows the Events for an object and links to the code executed when events occur. If necessary, new events and properties can be added to objects through inheritance.
- **Form Design area**, to create the application user interface
- **Code Editor**, to create event handlers.
 - **Database Explorer/Data Dictionary**, to browse database information.



Delphi 2.0's Object Repository centralizes object sharing for easily maintained code.

Rapid Application Development

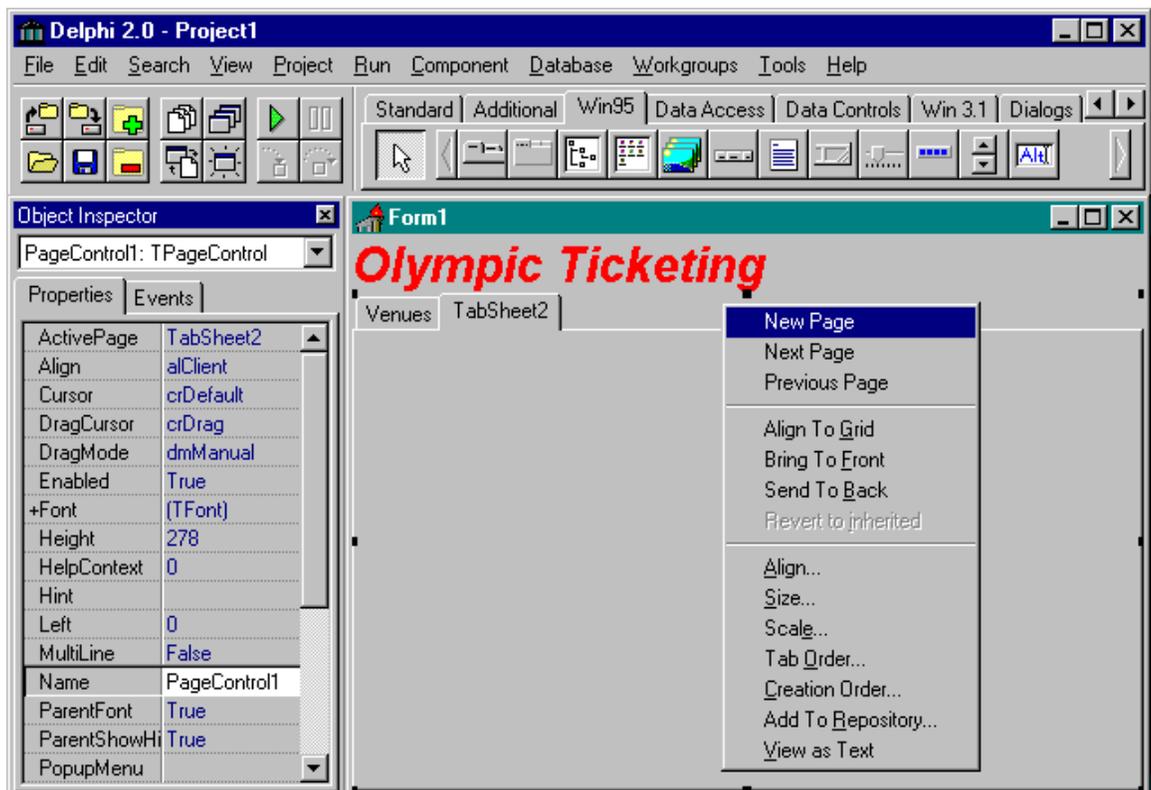
➔ Throughout this Quick Tour, you may follow along in Delphi 2.0 by executing the steps denoted by an arrow.

Delphi's object oriented architecture offers maximum reusability and maintainability - the result is shorter development cycles and easier to maintain code. Delphi 2.0 includes an enhanced implementation of the Visual Component Library.

To create an application, select an object from the Component Palette and drop it onto the form.

➔ From the standard page of the component palette, drop a caption on the form designer.

➔ Use the Object Inspector to modify the caption properties: enlarge the font. Change the Caption to *Olympic Ticketing*.



The Visual Component Library offers pre-built components for rapid application development.

One of Delphi's most important capabilities has been the full support for object-oriented programming. By having full support for encapsulation, polymorphism and inheritance in the Object Pascal language, Delphi has the unique capability of allowing developers to create their own custom objects, whether they subclass from existing visual components in Delphi or represent entirely new abstract business objects. And since Delphi is written in Delphi, there is no distinction between the types of objects a developer or third party could create and those

written by Borland. In fact, many third party vendors and Delphi developers have created a large and growing marketplace for these components.

Windows 95 User Interface

In addition to 32-bit support for long file names, multi-threading and the Windows 95 and Windows NT APIs, Delphi includes a complete suite of Windows 95 controls: TreeViews, Header controls, Status Bars, progress meters and more.

- ➔ Click on the Win95 tab page of the component palette, and drop a tabbed notebook.
- ➔ Right Click the tabbed notebook to add two new pages.

Because building on an existing foundation reduces development time, Delphi's *Object repository* stores forms, data modules and complete applications for reuse.

- ➔ Choose FILE | NEW from the menu, select a **Data Module** object from the available options.

Database Speed and Scalability

Delphi 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. Database access occurs through the core, 32-bit Borland Database Engine. The New Data Dictionary and Data Modules support business rule development. With centralized data integrity and business rules, developer's can respond quickly in dynamic business environments.

All data access for Delphi occurs through objects on the *Data Access* component page. Objects on the Data Access page encapsulate database source information about tables, queries and specific field references. These components work with the high performance Borland Database Engine to develop applications for Sybase, Microsoft SQL Server, Oracle, InterBase, Paradox, dBASE and the Local InterBase Server. 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.

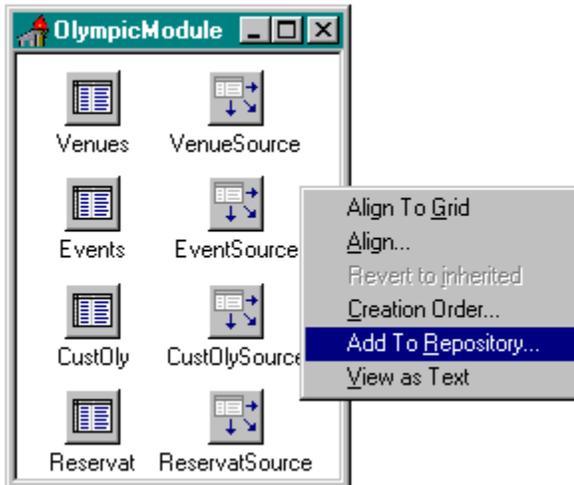
- ➔ Click on the *Data Access* component page. Drop 4 TTables and 4 TDataSource components into your data module.

Whether local or remote, an *alias*, or shortcut to your data, maintains a flexible pointer to where the data resides.

- ➔ Select each TTable and use the object inspector to set the DatabaseName property to DBDEMOS.
- ➔ From the Object Inspector, use the dropdown on the TableName property of a TTable to set the table to CustOly (the second to Events, the third to Venues and the fourth to Reservat) Change each TTables Name property to be more descriptive.

Delphi 2.0's new 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 business logic from a single location.



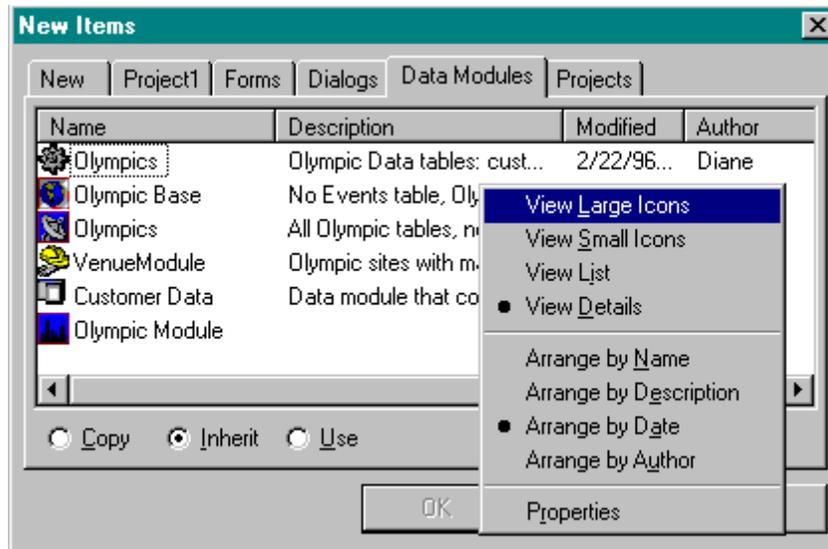
Data Module Objects centralize data access to easily implement business rules.

- ➔ Click in the Data Module and name it *OlympicModule*.
- ➔ Right click in the Data Module and select Add to Repository.
- ➔ When prompted, Save Unit 2 as OLYMODU.PAS

Once you have created a data module, it may be shared throughout an enterprise. All forms call into a central data module for display attributes and values. This 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. Developers can 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.

Object Repository

- Delphi 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. Any new application can inherit, use, or simply copy an existing structure – you pick the architecture that best fits your development needs.



Object Repository lets you copy, inherit or use any object for maximum productivity.

- ➔ Select the Data Module notebook page from the Object Repository, right click and View Details. Click on the *Modified* header so that the Data Module you just created appears at the top of the list.
- ➔ Press the Cancel button.

Drag and Drop database Development

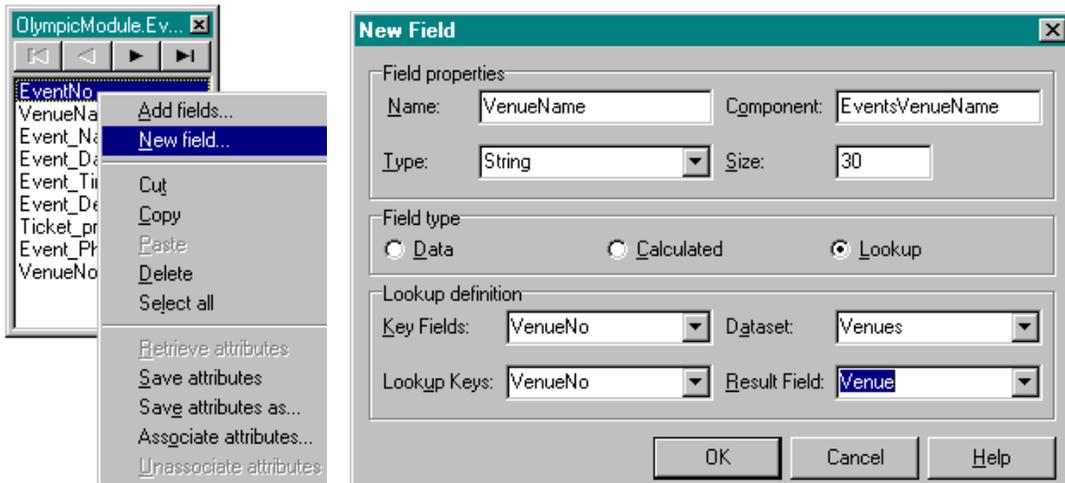
To enhance developer productivity, Delphi 2.0 identifies and eliminates many repetitive tasks.

- ➔ View your data module, OlympicModule (Shift F12 to locate)
- ➔ Double-click the Events Table and then right click to Add Fields. Click OK to add all selected fields.

Adding a Lookup

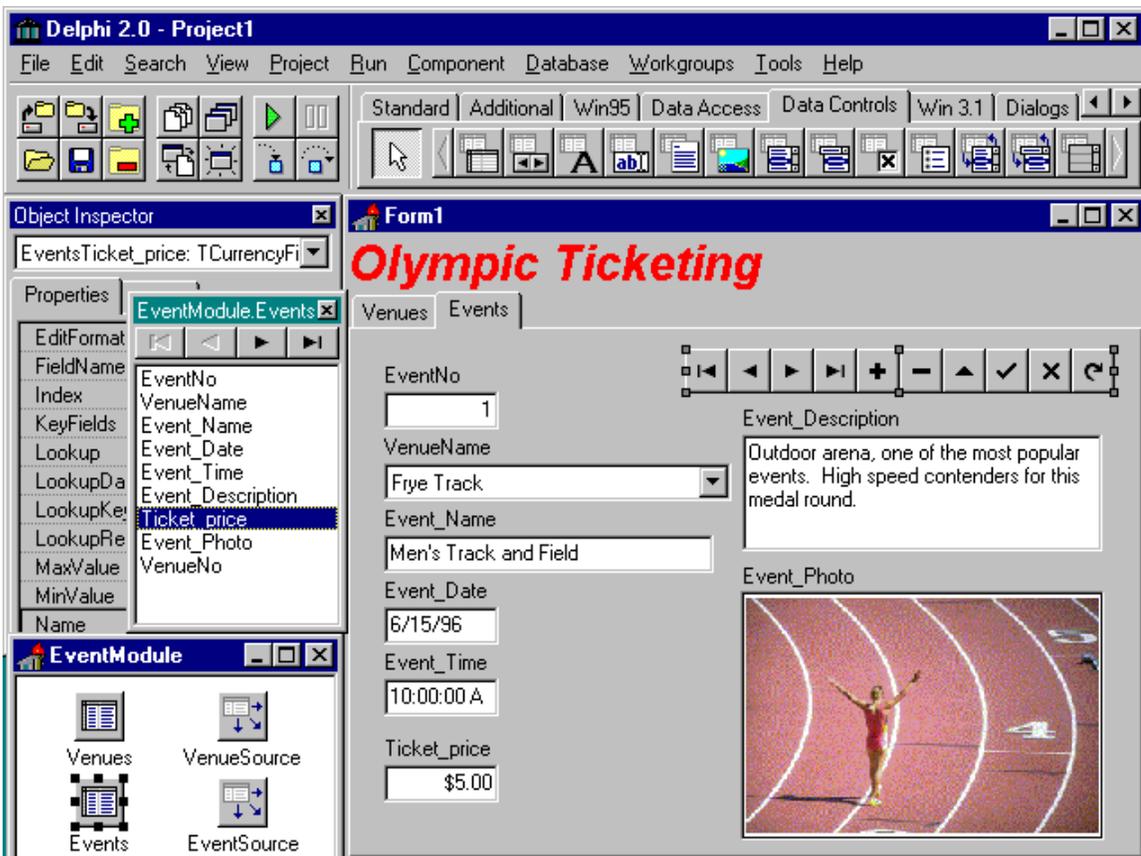
Database applications must seamlessly connect to a variety of corporate databases. Delphi's automatic lookups display data from multiple tables in a single source. The resulting application displays the data the users need in a format that is easy for developers to establish and maintain.

- ➔ Right-Click in the fields editor and choose New Field. Fill in the blanks as shown below.



Use a Lookup to seamlessly display data from multiple tables.

When you drag and drop fields from the Fields Editor, Delphi automatically formats display. Using the data dictionary, you can customize field objects and the data displayed within them.



Drag and Drop database development automatically formats the layout and display of multiple datatypes.

- ➔ Select all the fields, except Event_Description and Event_Photo and drag and drop them onto the tabbed notebook. Now select Event_Description and Event_Photo and drop them in the second column.

- ➔ Drop a DbNavigator from the Data Control page and connect its' DataSource Property to Events.

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 easily. Depending upon the event you select, the application will trigger and obey all attached code.

Compiler

Delphi uses an optimizing native code compiler to create stand alone executables (EXEs). The EXE is native machine code, instead of interpreted p-code, so applications run up to 15-50 times faster. Code execution speed becomes particularly critical when scaling applications up to large data sets where processing occurs against each record.

- ➔ Click the green arrow to compile and run your application.
- ➔ Use the Navigator to cursor through the records, press the + button to insert a new record, and the X button to cancel the record insert.

Visual Form Inheritance

Delphi 2.0 takes the fundamental OOP capabilities and extends them to the visual environment in order to make inheritance easier to use and more accessible. You can now visually inherit from forms while in the design environment, without writing code, and immediately see the effects of changes. For example, in many corporate environments it's desirable to create a standard "template" form, say for data entry, which will be used as the basis for many other forms. By using Visual Form Inheritance, you can be assured that as changes are made to the standard form they can be immediately inherited by other forms.

Visual Form Inheritance allows you to inherit all of the code, objects and properties with as many levels of inheritance as you like without runtime performance penalties. Other systems which attempt to implement a inheritance have severe performance penalties which render the feature unusable in the real world.



Delphi's Visual Form Inheritance allows you to easily create reusable, shared forms

- ➔ From the menu select File New, from the Object Repository, select the Project1 tab page and then select the Form 1 icon. An exact duplicate of Form 1 will appear in your design area, its caption will be Form 2.
- ➔ Move Form2 over and down. Position the two windows so that Form 1 is on top, but you can see both windows.
- ➔ Use the Object Inspector to change the Color property of the caption on Form 1. Drop a menu, double click to launch the menu editor, then right click to load a menu from a template.
- ➔ From one of the menu options on Form 1 enter the following code:

```
Form 2.showmodal;
```
- ➔ Switch to Form 2 and right click the tabbed notebook to add 2 new pages: Reservations and Customers.

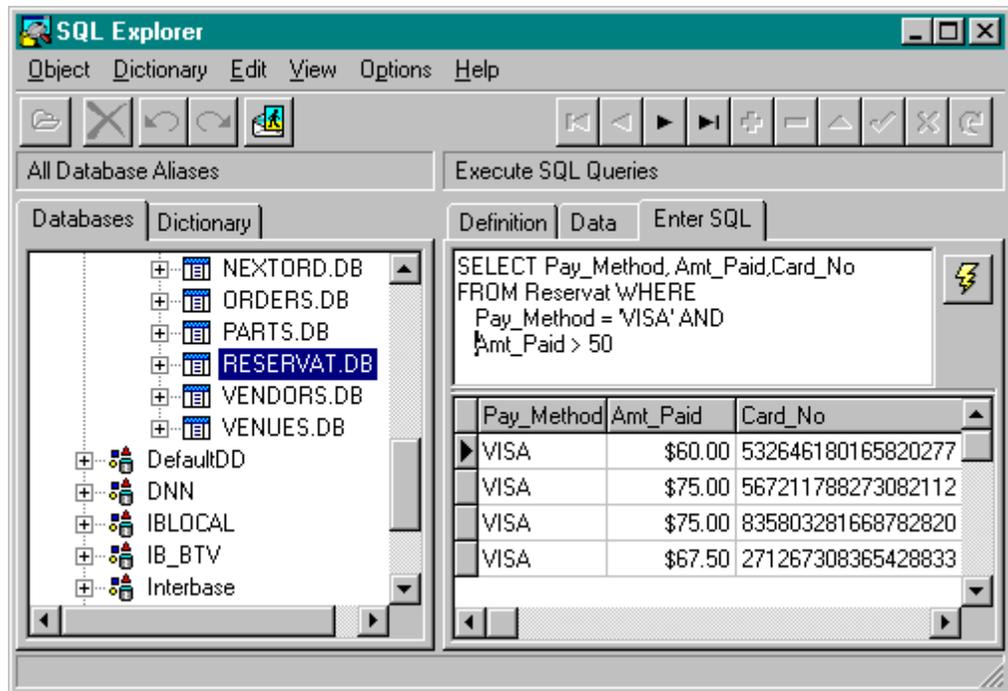
Developing a corporate standard in applications is important. Ensuring that these standards are adhered to is more difficult. Visual Form Inheritance and Form Linking extend object oriented programming to a visual paradigm ensuring that corporate and programmatic standards are maintained from project 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.

SQL Database Explorer

The *SQL Database Explorer* provides a graphical way of managing all your database demands, it supports browsing 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.



The SQL Explorer is an integrated tool for administering SQL and PC Databases

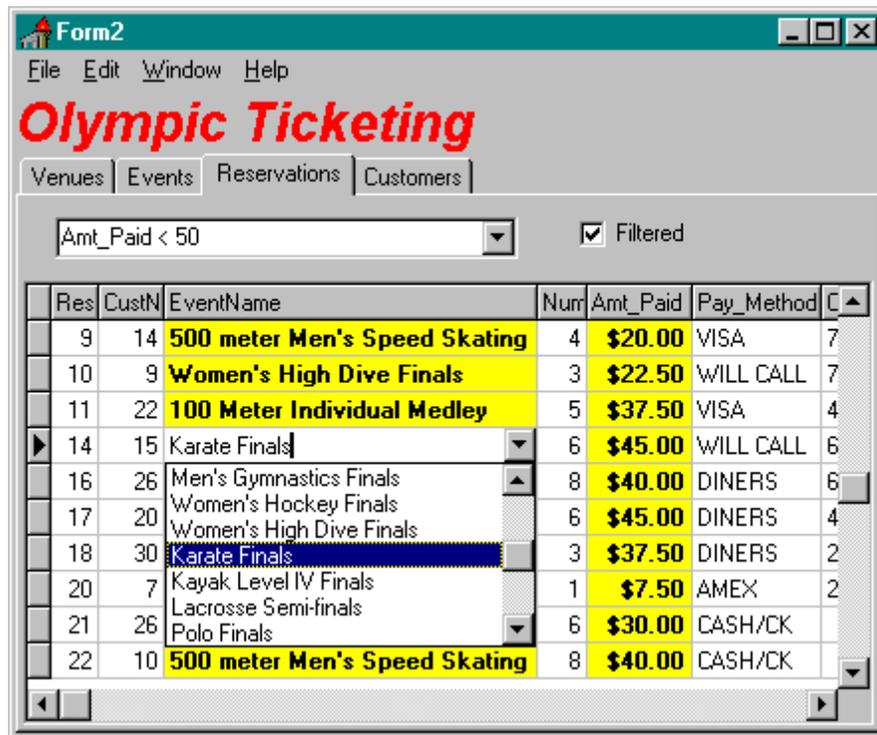
The SQL Explorer, also manages the *Scaleable Database Dictionary*. The simple to use interface enables the developer to easily define new domains of extended field attributes and then associate those to a field. The next time the field is used in an application, all the attributes are automatically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

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 create a data entry model that forces consistent data entry.

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.

- ➔ Double-Click to customize grid, add all fields then change the display attributes.
- ➔ The grid below has a lookup using EventNo from the Reservat table to the Events table to extract the Event field. (See Defining a Lookup.) The drop down appears automatically when a lookup is defined.



Expression Filters offer high performance filtered views of data sets while automatic lookups ease data entry in Delphi 2.0's custom grids.

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.

- ➔ On the Reservations Tab page, drop a Combobox and a Checkbox from the Standard component page.
- ➔ Double-click the Checkbox to enter the following code


```
OlympicModule.Reservat.Filtered := Checkbox1.checked;
OlympicModule.Reservat.Filter := Combobox1.Text;
```

- ➔ Select the Combobox and use the Object Inspector to change the TEXT property to
`Amt_Paid > 100`

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.

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.

- ➔ Click on the Customers Tabpage and then drop a dbLookupComboBox from the Data Control Component page.
- ➔ If you would like to synchronize a current customer with its orders, build a new datamodule with CustOly and Reservat. Set Reservat's MasterSource property to CustOly.
- ➔ Select the dbLookupComboBox and set the following property values:
 - ➔ Use the **ListSource** dropdown and choose CustOly
 - ➔ Type `CustNo;First_Name;Last_Name` into the **ListField** property
 - ➔ Set the **KeyField** property to 2 so you can search by last name, this property ties to the order of the fields in the ListField property.
 - ➔ Set **KeyFieldIndex** to CustNo, this is the value passed back when the user makes a selection
- ➔ Compile and test your application. From the menu choose the option to view Form 2, Click on the Reservation page and test the filter checkbox. Click on the Customer page, click in the dbLookupComboBox and begin typing the letters in the last name you'd like to search for.

DbLookupComboBoxes can act as instant search engines for records, or can lookup multiple fields from different tables. New Multi Object Grids display data in custom panels.

Multi-Object Grid

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.

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.

Reporting with Delphi 2.0

Delphi 2.0 supports two reporting solutions for maximum flexibility:

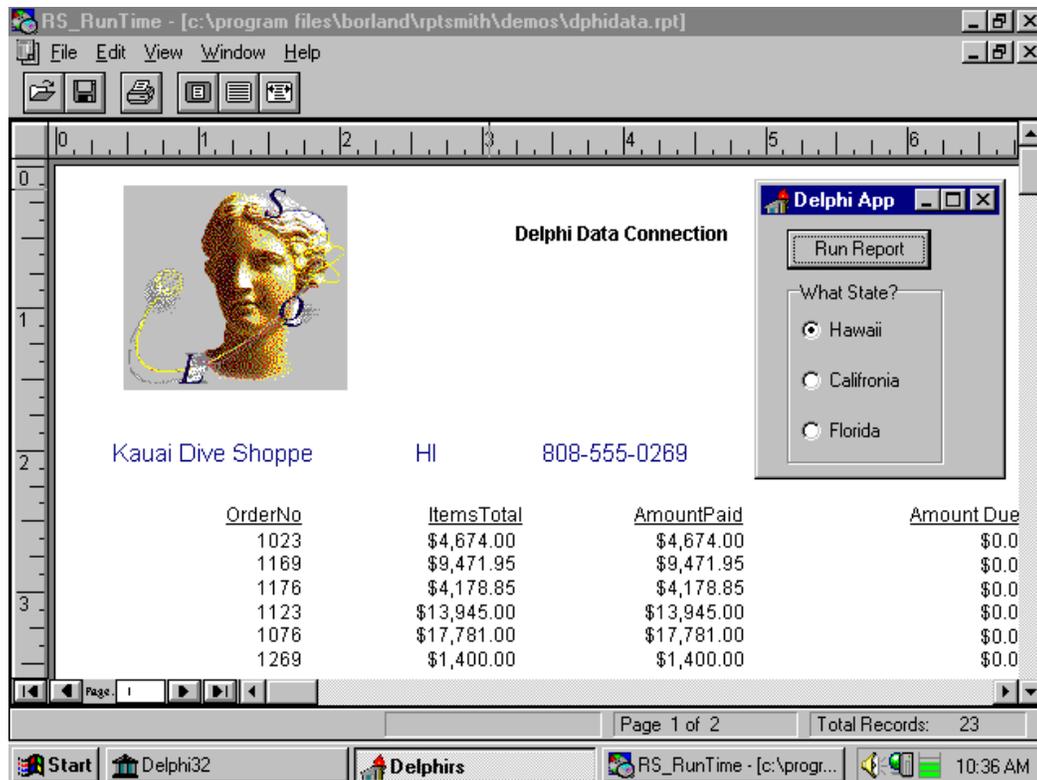
- **QuickReports**, for quick, simple, reporting with Delphi components. Available in all versions of Delphi.
- **ReportSmith**, for high volume, client/server report generation. Included in the Developer and Client/Server Suite.

ReportSmith

Borland's award-winning ReportSmith database reporting and query tool is integrated into the Delphi Environment. Reports are added to Delphi applications by selecting the Report component on the Database page of the Component Palette. ReportSmith includes support for:

- Prebuilt templates and styles
- Multi-level sorting
- Cross tab and mailing label reports
- Derived fields and summary values
- Custom group specifications

Reports can be distributed free of charge with the included ReportSmith runtime module. End users can purchase full versions of ReportSmith to create their own custom reports.

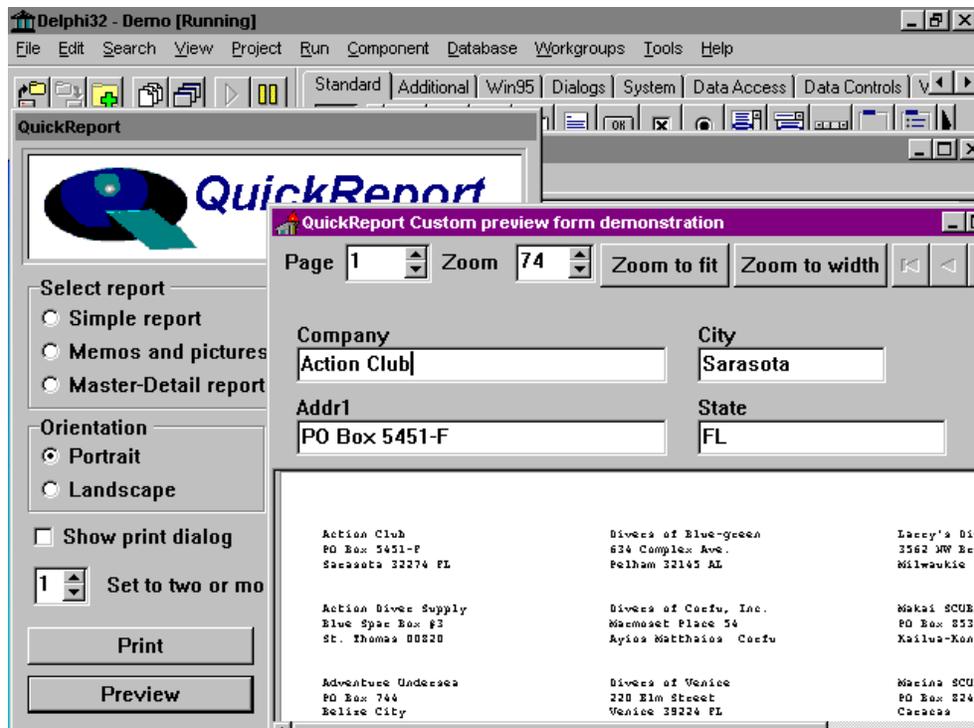


ReportSmith offers high volume Client/Server reporting.

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. 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.

QuickReports

All versions of Delphi includes QuickReports, an integrated set of Delphi components that rapidly creates columnar, master-detail, and label reports. While ReportSmith is optimized for large volumes of data that are typically found in corporate client/server applications, QuickReport is optimized for simpler reporting styles.



QuickReports integrates data reporting into Delphi 2.0 applications.

QuickReports 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.

Part Two: Technical Details

Part Two of the Reviewer's Guide studies some advanced topics in Delphi 2.0 including:

- Compiler Optimizations
- Application Management and Debugging
- Exception handling: The path to robust applications
- Visual Component Library - Building New Components
- Client/Server Solutions: Delphi's Database Architecture
- InterBase NT - Relational Database
- OLE Automation and OLE Controls (OCXs)

Compiler Optimizations

The new 32-bit native code compiler achieves its performance increase by using a number of new code optimization techniques. In the past, optimizing compilers often required experimentation with complex compiler directives to achieve the fastest performance. In addition, they often depended on performing additional “passes” over the code which slows down the compiler and discourages rapid application development. Delphi 2.0 uses many new optimization techniques automatically, without guesswork. In addition, the compiler *remains the fastest native code compiler in the world* performing at over 350,000 lines per minute on a Pentium. As a result, programmers always get the benefit of rapid application development and high performance.

The new 32-bit native code compiler includes automatic optimizations like

- *Register Optimizations* for faster access to variables.
- *Call Stack Overhead Elimination* to reduce function and procedure load time.
- *Common Subexpression Elimination* so that computations need only be run once.
- *Loop Induction Variables* to speed access to arrays and strings.

The result is faster performance. All of the optimizations performed in the 32-bit compiler are guaranteed correct and in no way change the meaning of the code. Delphi 2.0 also includes the capability of generating “Pentium-safe FDIV” code to guarantee that Delphi applications which use floating point division run correctly even on the so-called “flawed Pentium” processors.

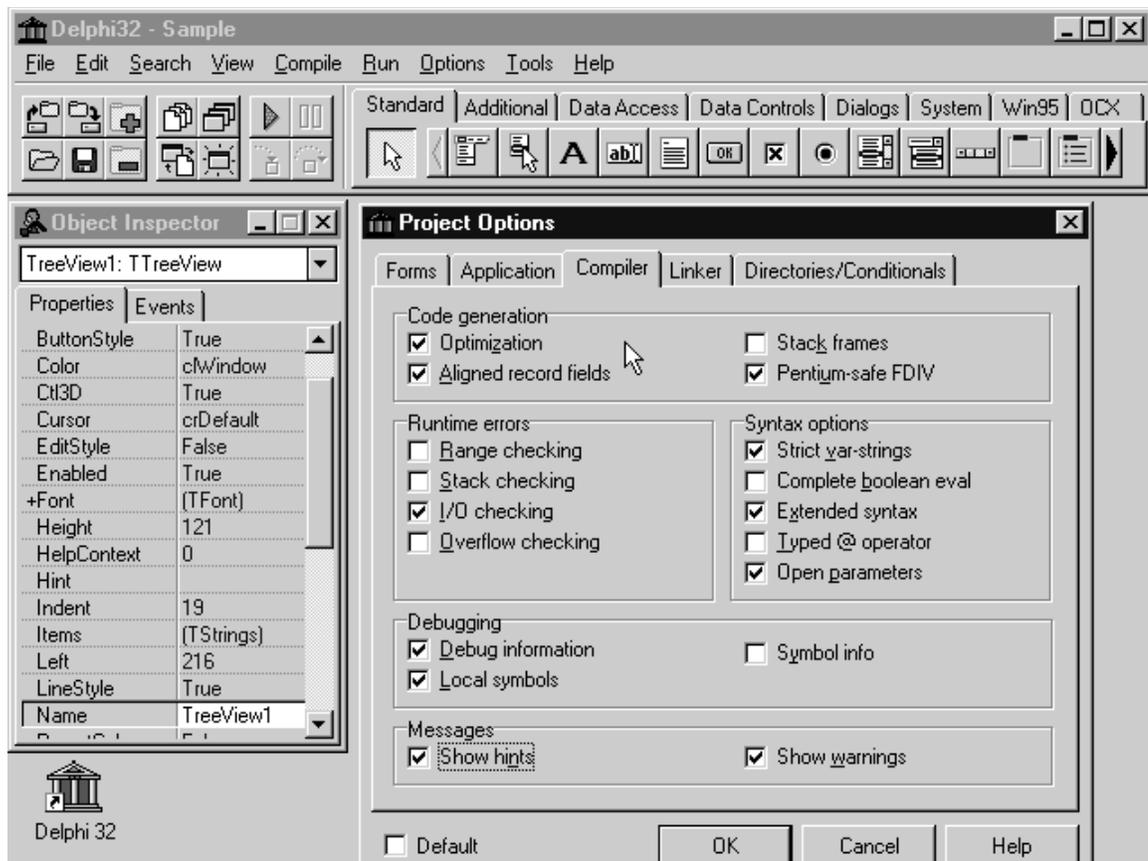
New Optimizing Linker

As part of the compilation process, Delphi uses a new 32-bit linking technology that also includes several optimizations to operate faster. The new linker is 20% to 50% faster due to a new unit caching scheme. This means that after the first time you compile an application, any forms or units that have not changed are

linked directly in memory rather than from disk. In addition, .EXEs are 20-25% smaller than before and still require no runtime interpreter DLLs.

Increased 32-Bit Capacity

The new 32-bit native code compiler runs in a 32-bit flat address space so that it completely eliminates all limitations previously associated with the 16-bit segmented architecture of Windows 3.1. For programmers this means it is now possible to take full advantage of all physical memory of the machine without resorting to direct Windows API calls. For example, you can declare arrays, strings, records and other data structures to be as large as you like, limited only by the operating system's limits. For example, on Windows 95 you can create strings up to 2 gigabytes. This is a great increase over the segment limitations of Delphi 1.0 applications on Windows 3.1 where data structures were limited to 64K.



Delphi 2.0's optimizing compiler creates native code that is up to 15-50x faster than p-code

New 32-Bit Data Types

The new 32-bit native code compiler also introduces several new data types to take advantage of the larger 32-bit flat address space, with additional flexibility and easy migration of 16-bit code. These new data types include long strings, wide strings, wide characters and variants. **Variants** provide the ability to change the type of a variable at runtime for more flexible database applications and OLE automation.

Enhanced statistical functionality for financial analysis includes:

- Support for a new currency type for **BCD arithmetic** for increased accuracy in financial applications
- **Math library** of common statistical functions like mean, standard deviation, and financial functions like net present value, interval rate of return and more.

Preliminary Benchmark results

Note: Further benchmark results for client/server application development are available in a separate whitepaper.

Benchmark tests show that code compiled with Delphi 2.0 applications can run from approximately 300% to nearly 400% faster than 16 bit Delphi applications. This means that Delphi continues to expand its performance advantage of over 10-20 times faster than p-code interpreters. For example, Delphi 2.0 Sieve benchmark results are 15 times faster than VB and 815 times faster than PowerBuilder. In addition, because of the new optimizing linker, .EXEs are 20-25% smaller than before and still require no runtime interpreter DLLs.

All benchmark tests were performed on a Gateway 2000 V66 (66Mhz 486 processor) with 16 megabytes of memory. The 16 bit benchmarks were performed using on Windows 3.1. The 32-bit benchmarks were performed with a pre-release version of Delphi 2.0.

Larger numbers indicate faster performance

loops / sec	Power Builder	Visual Basic	Delphi 16 bit	Delphi 2.0
Sieve	0.22	11.95	52.77	179.37
Whetstone	0.04	1.41	4.70	15.53
File write	0.05	0.42	0.74	2.89
File read	0.05	0.33	1.75	5.28

The Advantage of Native Code Compilers

Delphi always compiles immediately to optimized native machine code, unlike 4GL systems which sometimes allow you to generate C code which can be then compiled with a separate C code compiler. Although this sounds like an attractive way of correcting the major performance deficiencies of 4GL systems, in fact, Delphi has shown that there's a more efficient and reliable way to develop applications by always directly generating optimized native machine code. Because Delphi always produces optimized native machine code, it offers a number of advantages over a 2 stage "code generator" approach. These advantages include:

- **faster turnaround** -- by having the world's fastest native code compiler, Delphi increases productivity and encourages Rapid Application

Development. Unlike 2 stage “C code generators”, you never have to wait for the code generator or go through a separate generate / compile / link cycle.

- **easier testing** -- the code you work with in the development environment is the same compiled code you'll deploy in your production application. This means you don't have to worry about whether code that works with the interpreter will behave the same as the code created by the compiler.
- **high level debugging** -- since code that you write is compiled directly into machine code, the debugger lets you debug the code you wrote from within the environment. You never have to look at cryptic C code generated by the system. You never have to debug the C code created by the code generator.
- **easiest maintenance** -- you only have to maintain code in a single high-level language, object Pascal, rather than having to maintain code in both a 4GL language and in low-level C.
- **easier deployment** -- you can distribute your applications without runtime interpreter DLLs making it easy to deploy in the field.

Historically P-code systems and code generators have proven to be useful stop gap measures. For example, some of the early language implementations on the Apple II computer were implemented using the UCSD P-code system, which allowed programmers to work with high level languages such as Pascal instead of Basic on machines equipped with as little as 64K. However, since that time, many developers have been reluctant to use P-code systems due to their inherently slower performance.

Similarly, many of the first C++ language implementations were created as source code translators which translated C++ code into low-level C code which could then be compiled with any standard C compiler. However, the disadvantages of a slow turnaround time due to the two-stage compile process and the difficulty of debugging the generated C code, caused most C++ developers to seek “native code” C++ compilers instead.

Application Management and Debugging

Delphi has a powerful debugger that makes it easy to find and fix bugs in your code. You can set breakpoints, watch variables, and single step through code to understand its exact behavior.

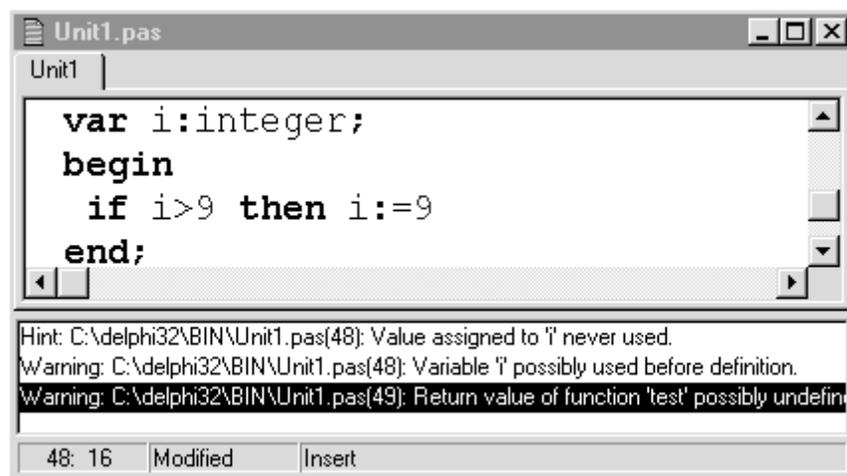
The environment is completely configurable so you can program the way you want to. You can customize editor keystrokes, the speedbar, the component palette, browser preferences, project defaults, and the color syntax highlighting. You can also add third-party tools or in-house utilities to the Tools menu and access system information through the Open Tools API. This makes it possible to completely integrate other tools into the environment.

Improved Compiler Error Messages and Diagnostics

One of the frequently overlooked advantages of a native code compiler is that it provides the programmer with a complete check of the program before running. Compilers can often catch logic errors that result from ambiguous or incorrect code that is not detected by an interpreter.

Delphi 2.0's new compiler architecture makes it easier to write correct code with:

- **Multi-error architecture** that finds all errors, making verification of large programs easier.
- **New hints and warnings** that detect common coding errors like unused variables, type mismatches and uninitialized variables or pointers.
- **Improved error messages** for a much clearer indication of a problem. (For example, a forgotten semi-colon.)
- **Smart module management** with File Uses and form linking.
- **Automatic form linking** to talk across various forms and data modules.



The screenshot shows a window titled 'Unit1.pas' containing the following Pascal code:

```
var i:integer;  
begin  
  if i>9 then i:=9  
end;
```

Below the code, the compiler messages pane displays the following:

```
Hint: C:\delphi32\BIN\Unit1.pas(48): Value assigned to 'i' never used.  
Warning: C:\delphi32\BIN\Unit1.pas(48): Variable 'i' possibly used before definition.  
Warning: C:\delphi32\BIN\Unit1.pas(49): Return value of function 'test' possibly undefin
```

The status bar at the bottom shows '48: 16 Modified Insert'.

Delphi 2.0 provides improved error messages and diagnostics.



For more information on compiler optimizations, see *An Overview of the Delphi 2.0 Optimizing Native Code Compiler for Windows 95 and NT*.

Reuse and Maintainability

Delphi's component architecture is efficient and flexible. Many systems which provide an easy visual development environment have inherent limitations that get in the way of advanced developers who need power. Because Delphi's architecture is completely component based, there are no such limitations. As proof of the power of the environment, Delphi is written in itself, using the same library of components and tools that are available to developers.

Delphi's object-oriented programming support shortens development cycles by offering maximum reusability and maintainability. Delphi is a completely object oriented programming paradigm with support for polymorphism, encapsulation and inheritance.

Object Pascal

Delphi's programming language, Object Pascal, is high-performance, structured, and object-oriented. Object Pascal meets the needs of today's developers by providing the ease of development of a 4GL with the performance and flexibility of a 3GL. For programmers who have already learned structured programming techniques, whether it's in C, Basic, Pascal, COBOL, FORTRAN, dBASE, Paradox or any other language, Object Pascal provides a natural migration path offering maximum productivity without complexity. Object Pascal leverages your current skill set and requires little education time. As the underlying foundation with which applications are built, Object Pascal is a key strength of the Delphi environment.

Object Pascal has built-in support for component-based development, making it easy to develop Windows client/server applications without requiring knowledge of the Windows message architecture, use of pointers or memory allocation. And it provides the full power of a fully compiled, object-oriented language. So you have the power to create reusable objects and dynamic data structures such as linked lists and pointers. You can also create reusable Dynamic Link Libraries (DLLs) and separately compiled libraries.

Object Pascal has full support for object-oriented concepts including encapsulation, inheritance and polymorphism and has been extended to include advanced facilities such as properties, privacy, runtime type information, exception handling, delegation and class references.

Delphi also supports such low-level programming facilities as:

- Subclassing of Windows controls
- Overriding the Windows message loop
- Handling any Windows message
- In-line assembly language code

Complete details of object-oriented programming in Delphi are covered in the *Delphi User's Guide*.

Visual Component Library - Building New Components

The components used in Delphi are built upon an application framework, or a set of object types used to build the foundation of an application. This framework is called the Visual Component Library (VCL), and uses the object model implemented in Delphi to give developers a true object-oriented development system.

VCL32 includes full encapsulation of the Windows 95 user interface elements (Tree Views, Trackbars, sliders, progress bars, Toolbars, Rich Edit, List Views, Image Lists header and status bar controls, etc), along with many additional components. Out of the box, VCL32 includes most of the standard components used by Windows programmers such as user-interface objects for editing input, combo boxes, and list boxes, grid, tab and notebook objects.

A key feature of Delphi is the ability to not only *use* components in a visual design environment, but to *create* new components as necessary. New components, through the inheritance feature of object-oriented programming, can be as simple as an existing component with some added functionality, or a completely custom component that is based entirely on new code.

The ability to create new components means developers don't have to switch to another development environment when new, custom components are required. Designing and implementing new components in Delphi is a straightforward process that allows developers to change the defaults of a given control, give existing controls some new functionality, encapsulate existing third-party Windows control, or create components that do something completely new.

There are three steps to creating a new Delphi component:

- inherit from an existing component type
- define new fields, properties and methods
- register the new component

The example code below shows a simple visual component which is a new type of button called `TCmdBtn` that inherits from the `TBitBtn` class. This button type has a field called `FCommand` that contains a command string, which can launch any Windows application. When the user clicks on the button at runtime, the contents of the command string will be executed by calling the Windows `ShellExecute` function.

This example also shows how runtime type information is used to provide automatic display and editing of properties in the Object Inspector. By defining a published property called `Command`, the user can automatically edit the value of the corresponding `FCommand` field. In this case, assignments to the property are simply passed through to the `FCommand` field. In other cases, we could define “set” and “get” routines that provide data validation or limit the access to

the field. You can also define properties which are read-only and cannot be modified.

Because the `TCmdBtn` inherits from `TBitBtn` it automatically gets all of the properties for that object type, including its icon on the palette. In addition, it also automatically inherits the display of a pop-up help hint on the palette, object persistence and default naming of objects in the Delphi environment. The object that we created is no different from the objects supplied with Delphi. In fact, these are the same coding techniques that were used to build the standard components in Delphi itself.

```
type
  TMyButton = class(TSpeedButton)
  private
    { Private declarations }
    fcommand: string;
  protected
    { Protected declarations }
  public
    { Public declarations }
    procedure click; override;
  published
    { Published declarations }
    property command: string read fcommand write fcommand;
  end;
procedure Register;
implementation
procedure Register;
begin
  RegisterComponents('Samples', [TMyButton]);
end;
procedure TMyButton.Click;
  var
    buffer: array[0..255] of char;
  begin
    Winexec(strPCopy(Buffer, Fcommand), SW_Show);
end;
end.
```

We're calling the "WinExec" function in the Windows API. This function is readily available. I don't have to write a declaration, figure out what the parameters and specify what DLL it comes from. The Windows API is built into the Delphi libraries. Add the component to the palette by selecting Component | New from the menu.

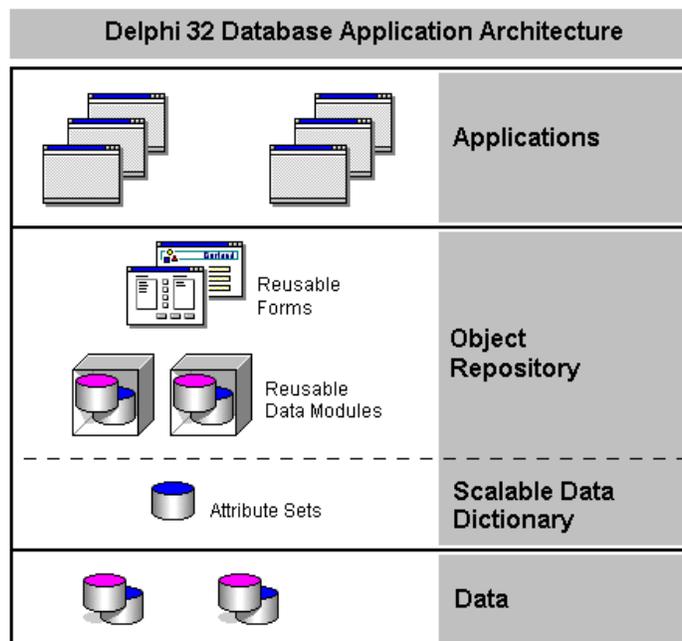
And that's it! We now have a functional component, ready for reuse.

Client/Server Solutions: Delphi's Database Architecture

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*

A Delphi database application is built using data-access components and data-aware user interface components. Delphi components communicate with the Borland Database Engine, extract information from the Data Dictionary, and obey business rules set up on the server or in the client.



The object oriented architecture supports the development of a reusable set of Objects, Business Rules and Forms. The **Data Module**, **Scaleable Database Dictionary** and **Object Repository** are technologies that become the framework for the architecture. This architecture allows you to separate GUI, Business Logic and Database design to reduce development time and improve maintainability

Borland Database Engine

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 Link 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.

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

- 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 scaleability. 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 SQL Link Drivers** with SQL 92 compliant access to Oracle, MS SQL Server, Sybase System10, Sybase System4, 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** that 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.

Logical Application Distribution:

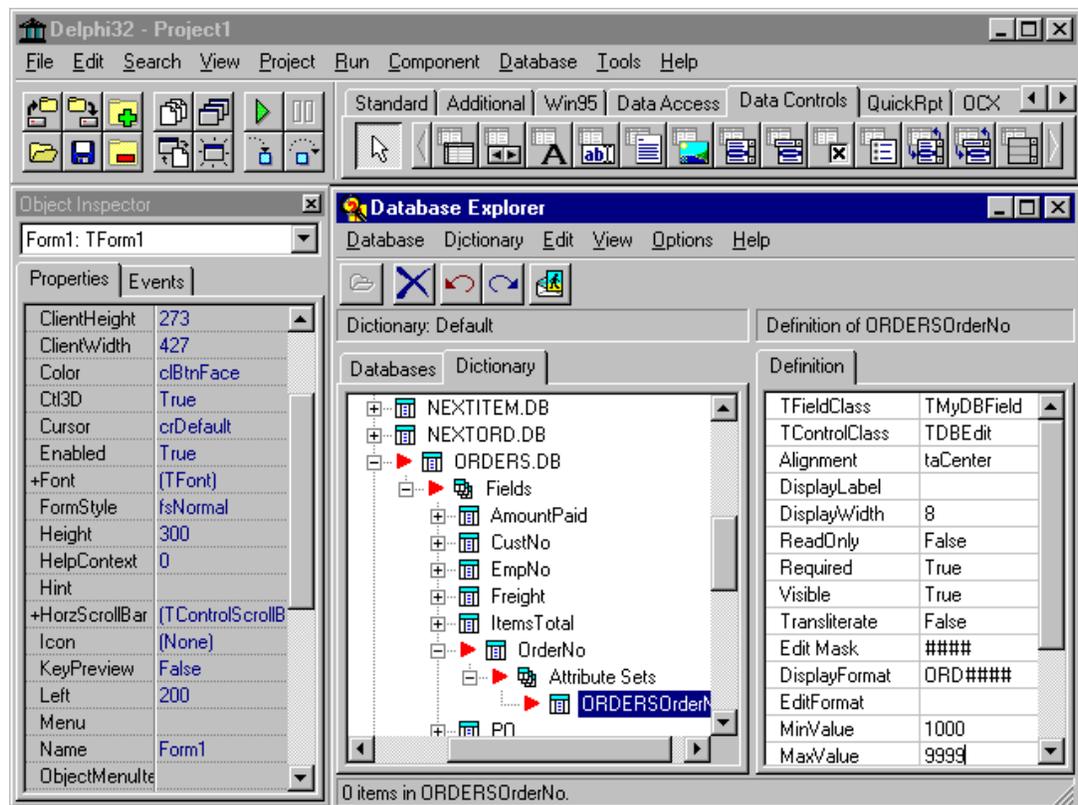
The Data Module allows developers to create a logical n-tier application environment by centrally locating business logic and separating it from the GUI and database design. This is a solid foundation for physical support of distributed applications. Support for physical n-tier application and server 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.

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 maintains consistency in data and reduces network traffic.

Using the data dictionary has two advantages:

- **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.
- **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.

The data dictionary is ALIAS based, transportable to any server. Aliases are your link to database locations. They increase scalability by giving you a single location to define database access and they let you avoid typing long path names. Open the Database Explorer and click on the Dictionary page. The Database Explorer page lets you browse all your server meta data, table structures, and data. The Dictionary page lets you set up extended field attributes.

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 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.

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.

- **Navigational Updates** are a mechanism by which each local transaction is sent individually to a database server within the context of a SQL Transaction.
- **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.

Cached Updates

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. 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. By not updating the underlying tables directly, record locks and contentions are minimized resulting in shorter wait times by end-users. Because users communicate with the server less frequently, Cached Update reduces the amount of network traffic that may slow the application.

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.

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.



For technical details about building database applications, see the *Delphi Database Architecture Whitepaper*.

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 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.

Exception handling: The path to robust applications

To make your applications robust, your code needs to recognize and respond to user exceptions, database exceptions, and other system errors. So that applications can recover without losing or corrupting data, Delphi provides you with a mechanism to handle errors in a consistent manner. 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, code loss and frustrated users.

Delphi offers developers C++-style exception handling to handle errors. Exceptions are represented as objects that contain specific information about a

particular error (such as the type and location of the error). 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.

Implementing Exception Handling

Exception handling is implemented using exception-handling blocks (also called protected blocks) that are set off with the keywords **try** and **end**. There are two types of protected blocks, **try..except** and **try..finally**.

try..except

This block is used to handle exceptions. The general syntax looks like this:

```
try
    {statements to be executed}
except
    {responses to exceptions}
    on exception1 do statement;
    on exception2 do statement;
else
    {default exception-handling code}
end;
```

Should any of the statements to be executed within the **try** part of the block raise an exception, execution jumps to the **except** part. If the exception has been anticipated and is listed among the **on..do** statements, the corresponding code is executed and the block is exited, otherwise the statements in the **else** part of the block are executed and the block is exited.

try..finally

Whereas **try..except** is intended to facilitate recovery from exceptions, **try..finally** is designed to make sure that specific code (typically, cleanup code) is executed before anything else happens. The general syntax looks like this:

```
try
    {statements to be executed}
finally
    {cleanup code}
end;
```

The exception handling in Delphi gives developers an easy way to anticipate exceptions, using a syntax that's easy to understand. The ultimate payoff, however, is the ability to build robust applications. Developers can handle and recover from serious systems faults, like General Protection Faults (GPF on Win3.1), or Access Violations on Win95. Delphi eliminates the instabilities frequently found in other visual development environments.

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.

Local InterBase Server

Both Delphi Developer and Delphi Client/Server Suite include a version of InterBase 4.0 that runs as a local high-performance server on Windows 95/NT. Since InterBase is an ANSI SQL-92 compatible database management system, this ensures complete scaleability with other ANSI compatible database managers including Oracle, Sybase, Informix and InterBase running on Intel or Unix platforms. The Local InterBase Server also ensures that you can do

efficient “off line” development for testing purposes or when the remote server is unavailable. The Local InterBase Server fully supports a wide range of advanced features including:

- ANSI 92 compatibility
- Computed fields
- Outer joins and join expressions in the From clause
- Unions in the Select statement
- Complex Data including Binary Large Objects and Multi-dimensional arrays
- Advanced features including stored procedures, triggers, constraints
- Simple installation and maintenance

OLE Automation and OLE Controls (OCXs)

Delphi 2.0 includes complete support for OLE facilities on Windows 95 and NT such as:

- the ability to create OLE automation controller and server applications
- full compatibility with forthcoming Network OLE and Remote Automation for partitioning
- the ability to use existing third party OLE controls (OCXs)
- the ability to customize OCXs via inheritance

The Importance of being OLE

Microsoft's OLE technology includes a variety of important capabilities for increasing the ability of developers to create more modular and integrated applications. The goal with Delphi 2.0 has been complete adherence to Microsoft system standards to ensure that developers can use Delphi 2.0 to create a diverse range of applications without limitations. Delphi 2.0 goes beyond simply adhering to Microsoft standards; it actually makes the use of OLE technology easier through object orientation. As a result, OLE technology is completely integrated into Delphi 2.0 and ensures compatibility with future technologies such as Network OLE.

The OLE support in Delphi 2.0 includes the ability to easily install and use OLE controls (OCXs) as well as the ability to easily create OLE automation controllers and servers. For maximum flexibility, Delphi 2.0 can create both in-process and out-of-process servers. By supporting both OLE automation controllers and servers, Delphi 2.0 is completely compatible with the forthcoming Network OLE technology as well as VB 4.0's remote automation technology, with the added advantage of faster performance. In addition, because Delphi 2.0 is a native code compiler, advanced developers are able to write their own OLE controls (OCXs) within Delphi 2.0 though this is more difficult than creating Delphi components.

Using OLE Automation

Delphi 2.0 makes use of a new type, called **variant**, to provide seamless integration of OLE automation. With equal ease, Delphi 2.0 allows you to create applications which can be either an OLE automation controller or an OLE automation server. An OLE automation controller is the most common use of OLE automation among application developers and system integrators. For example, a Delphi 2.0 application can be used to control another OLE application, such as Word, Excel, Paradox, Quattro Pro and others.

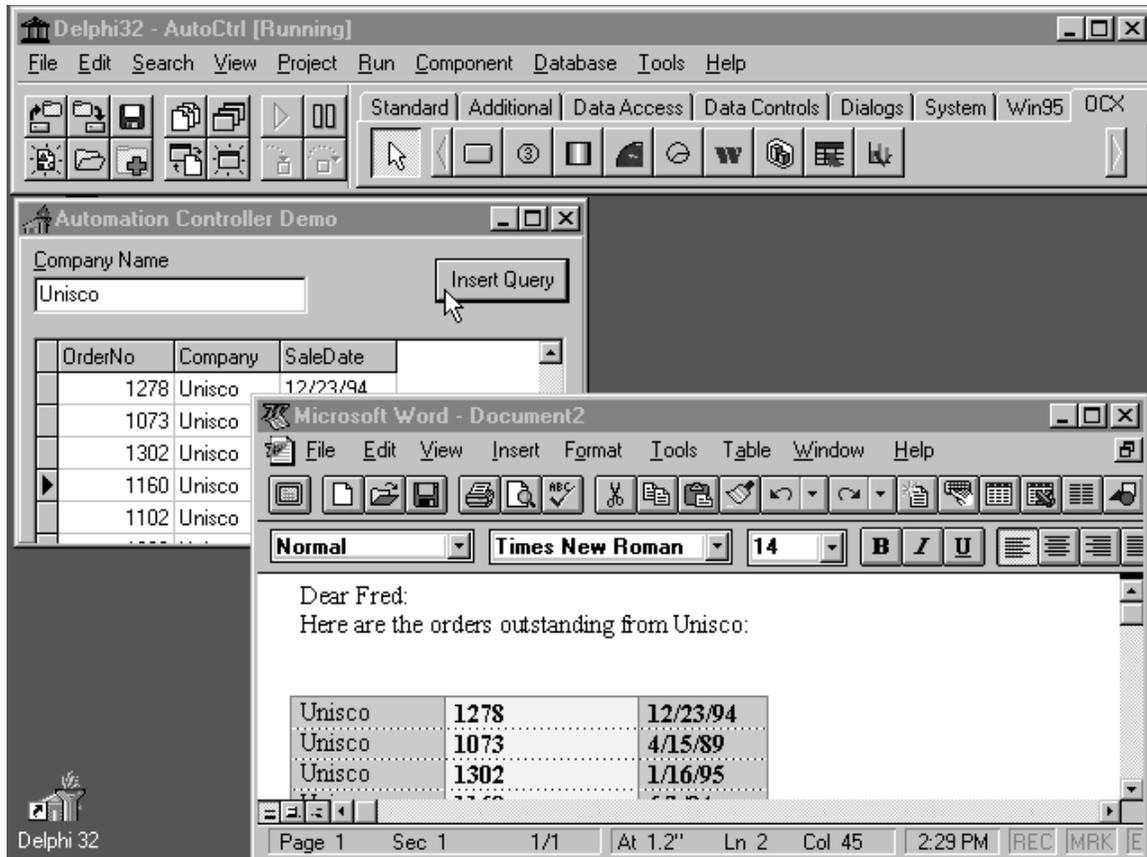
The variant type allows developers to declare variables whose type is determined at runtime, allowing them to take advantage of the inherent flexibility of OLE automation. In effect, you can use a single variable to connect to different types of OLE automation servers at runtime. Delphi 2.0 also introduces the ability to use named parameters when making calls to OLE automation servers. For complex functions which often have dozens of parameters, developers can

simply supply the parameters of interest and use the server's default values for the rest.

Office Integration

The code below is from a sample Delphi 2.0 application which performs a query and then inserts the result set into a Word document. Note that the OLE automation takes only four lines of code in addition to the declaration of the variable MSWord of type variant.

```
{ This example uses OLE automation to insert a query result
into Word}
procedure TForm1.InsertBtnClick(Sender: TObject);
var
  MSWord: Variant;
  S: string;
  L: Integer;
begin
  { Connect to the automation server in MS Word and run the
query}
  MSWord := CreateOleObject('Word.Basic');
  with Query1 do
    begin
      Close;
      Params[0].Text := Edit1.Text;
      Open;
      try
        First;
        L := 0;
        while not EOF do
          { Store the query result set in string S }
          begin
            S := S + Query1Company.AsString + ',' +
              Query1OrderNo.AsString + ',' +
              Query1SaleDate.AsString + #13;
            Inc(L);
            Next;
          end;
        { Use OLE automation to insert S into the Word document }
        MSWord.Insert(S);
        MSWord.LineUp(L, 1);
        MSWord.TextToTable(ConvertFrom := 2, NumColumns :=
3);
      finally
        Close;
      end;
    end;
  end;
end;
```



Use Delphi 2.0 to create OLE automation controllers and servers.

Creating OLE Automation Servers

Delphi 2.0 also enables you to create your own OLE automation servers. These can be either in-process or out-of-process (or local) servers. You can expose functions or methods of your application so that they can be called from other applications such as Microsoft Word, Excel, Visual Basic, C++, Paradox and Delphi 2.0. Because Delphi 2.0 can produce both OLE automation controllers and servers that are highly optimized native code executables, it offers a unique performance advantage that will be increasingly important with the emergence of Network OLE. Delphi 2.0 developers will be able to take advantage of having applications that are partitioned and have the fastest compiled code running on both the client and the server side.

To create a new OLE automation server you can use the Automation Object Expert. The expert automatically defines a new object that inherits from the TAutoObject type and sets up all of the OLE registration for you including the program ID, class ID and instancing options.



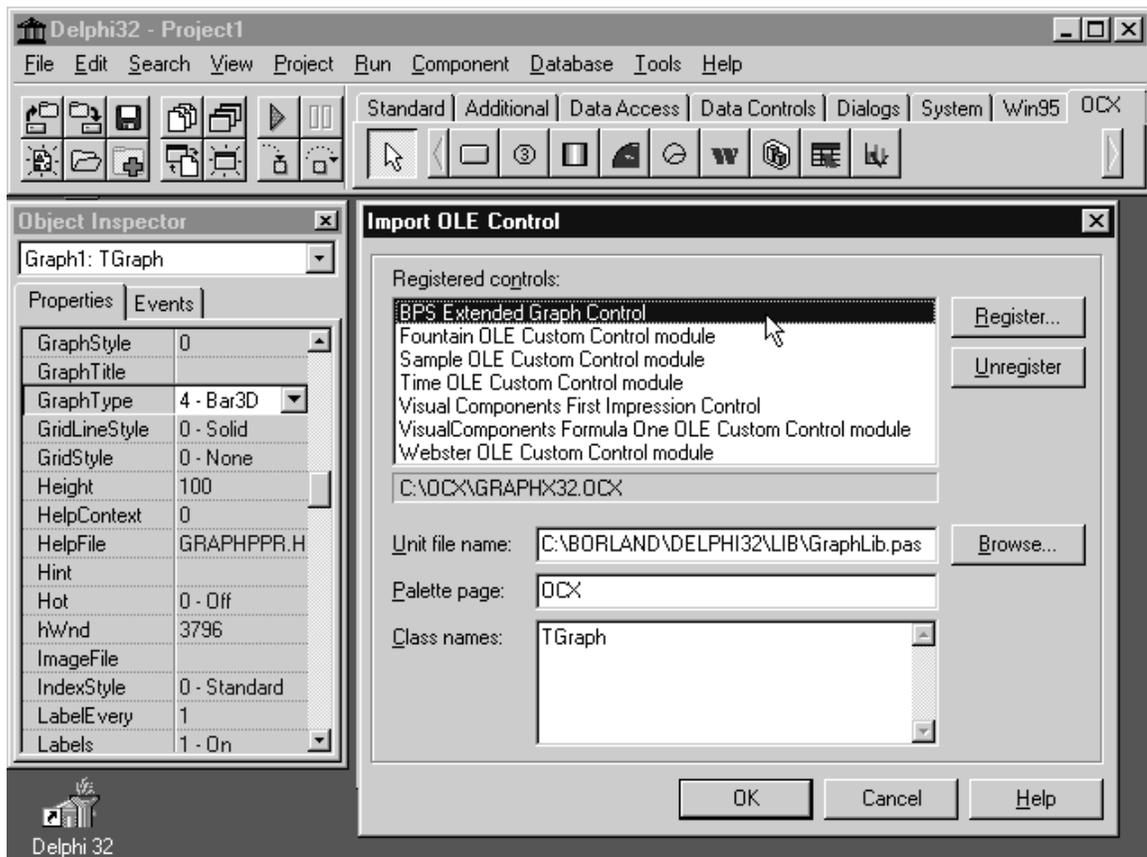
For more information about building OLE Automation Servers, the Delphi Users Guide.

Using OLE Controls (OCXs)

Because Delphi 2.0 is a completely object-oriented environment, integration of OLE controls is seamless. You can install third party OLE controls (OCXs) just

as you can install components you write in the Delphi environment. Delphi 2.0 gives you complete access to the OLE system registry so that you can load OLE controls and register them in one simple dialog.

When you install an OLE control (OCX), Delphi automatically creates an object wrapper to provide a completely object-oriented view of the control. Delphi 2.0 is the only rapid application development environment that makes OLE controls completely object-oriented. ***This makes it possible for developers to easily subclass any OCX, so that they can be easily customized through inheritance.*** An object wrapper is automatically generated by Delphi when you install any OCX.



Delphi 2.0 lets you easily install OLE controls as well as your own Delphi components

Once an OLE control is installed into the Delphi 2.0 component palette, it can be used just like any of the supplied Delphi components. All of the OLE controls properties and events are fully accessible within the environment through the Object Inspector.

Conclusion

As you try Delphi, you'll find that it offers a unique combination of high-performance, code reuse, rapid application development and database scalability. Its unique native code compiler technology and scalable database architecture allow developers to breakthrough the performance limitations to deploy production quality workgroup and enterprise client/server applications.

Delphi 2.0 is a totally new product built from the ground up on an optimizing compiler to take advantage of the improved performance of 32-bit platforms including Windows 95 and Windows NT. There are many new compiler optimizations, linker optimizations and new 32-bit data types so that the performance advantage of Delphi applications over p-code interpreters is even greater than before. In addition, the new compiler architecture offers easy access to important system features such as multi-threading, OLE automation and OLE controls (OCXs).

As business challenges continue to grow, Delphi advances its robust object-oriented architecture to meet the needs of application developers. Its flexibility and responsiveness to customer needs continues to grow.

Appendices

Documentation Overview

The documentation included in Delphi Desktop:

- Getting Started
- Delphi User's Guide
- Delphi Component Writer's Guide
- Delphi Database Application Developer's Guide
- Object Pascal Language Reference

Delphi Developer also includes:

- Reference Library Guide
- ReportSmith Creating Reports
- InterBase Getting Started

Delphi Client/Server Suite also includes:

- InterBase Language Reference
- InterBase Data Definition
- SQL Links User's Guide

Hardware/software requirements

The Delphi Client/Server Suite 2.0 requires:

- Windows 95 or Windows NT
- 30 megabytes of disk space for a minimum install
- an 80486 or faster processor
- 8 megabytes of RAM

To install, run the INSTALL.EXE program from the supplied CD ROM or floppy disk set and follow all instructions. Additional last minute information is included in the README.TXT file normally installed in the \DELPHI32 directory

Many sample applications are included in the \DELPHI32\DEMOS directory.