

Borland[®] Delphi[™] 2005 overview

The complete development solution
for Microsoft[®] Windows[®]

A Borland White Paper

By Jeremy McGee, Bassett Data Systems

October 2004

Borland[®]

Contents

Delphi 2005: The leading independent development solution for Windows	3
The importance of independence	3
All the languages, technologies you need	5
Delphi for 32-bit Windows.....	5
C# or Delphi for the .NET Framework.....	6
Delphi: The language evolves and grows	7
Streamlined environment for fast code development.....	8
Refactoring and beyond: Outstanding code productivity.....	11
Guaranteed code quality with integrated unit tests	13
Performance, baked in from the start	14
Check out, refactor, unit test, check in: Team development, simplified	15
Better technology for faster development	16
Delphi RAD for ADO.NET	16
Distributed data with DataSync and DataHub resolver and provider	17
The Data Migration Wizard.....	18
Enhanced and upgraded Win32 database support, and easy migration to .NET.....	20
Simple development and deployment of Web applications	21
Streamlined Web forms for ASP.NET	21
Atozed IntraWeb: Fast RAD Web form design for Win32 and .NET	22
Deploy Web applications easily, quickly, and accurately.....	22
Beyond databases: Model-powered development with Borland® Enterprise Core Objects (ECO™) II.....	23
Beyond Windows: Integration with other systems.....	26
Delphi 2005: The broadest solution for Windows development	27

Delphi 2005: The leading independent development solution for Windows

Delphi 2005 is the ultimate and complete development solution for Microsoft® Windows.® Combining the power of the Delphi and C# languages with the performance of the Microsoft® .NET Framework and Win32,® Delphi 2005 delivers a hyperproductive RAD development environment.

With more productivity-enhancing features for designing and communicating architecture, developing Web, database, and rich-client applications and a seamless framework for application lifecycle management, Delphi 2005 equips Windows developers with the solution they need to deliver better software, faster.

The importance of independence

As enterprises increase their reliance on internal software development, developers must create more applications faster than ever—with the added demand that applications be deployed more widely across the enterprise.

Windows developers face particular challenges when they deliver these new projects:

- Demand for new, faster, and more flexible development paradigms such as model-driven applications and service-oriented architectures are not supported by existing development solutions.
- Many enterprises employ a mix of databases and application servers. Frequently several programming languages, such as Microsoft® Visual Basic,® C#, Java,™ and Delphi are in use, and development is in transition to the Microsoft .NET Framework.
- Existing Windows development tools, which focus purely on the individual programming task, do little to optimize communication between development team members and roles.

- Adapting systems to match changing user requirements can be difficult when code must be manually refactored; and without integrated unit testing, ensuring that code quality is maintained during changes presents a challenge.
- Delphi 2005 provides a complete solution for Windows development that addresses each of these issues by combining the most popular Windows languages and SDKs in one easy-to-use environment.

The solution includes a broad range of Windows technologies that dramatically speed development of rich-client applications, Web applications, Web Services, and custom components. These include:

- A powerful multilanguage, multiplatform IDE with integrated refactoring, error display, in-line documentation viewing, and application profiling.
- Integrated access to Borland® StarTeam®, enabling teams to work simultaneously on the same part of an application without holdups
- Delphi RAD for ADO.NET to radically speed and simplify development. Improved drivers and components with ADO.NET database access, including remoting heterogenous database access and enhanced data-driven Web components to make ASP.NET applications easier to develop
- Borland® Enterprise Core Objects (ECO™) II, a highly scalable model-powered application framework that offers services such as object-relational mapping, object caching, and object presentation, all powered by Unified Modeling Language™ (UML®) models, making enterprise .NET development faster, more reliable, and more maintainable
- Borland® Janeva™ integration, enabling Windows developers to work seamlessly with Java application servers

Because these are all available directly from Delphi 2005, you'll have no worries about incompatibilities or any need to switch from one program to another.

Delphi 2005 has the richest set of capabilities for Windows development. Delphi 2005 provides everything needed to increase Windows developer productivity across the enterprise.

All the languages, technologies you need

Delphi 2005 is unique among Windows development solutions in its ability to target both Win32 and the Microsoft .NET Framework.



Figure 1: Choose ASP.NET, Windows Forms, Delphi, or C# from one IDE

Delphi for 32-bit Windows

For more than 10 years, the 32-bit Windows API, Win32, has dominated client application development. A wealth of experience and code exist for this API, which, understandably, many development organizations are reluctant to give up.

Delphi 2005 provides a comprehensive, advanced, and enhanced solution for Delphi language programmers targeting Win32. A broad range of components provided with the Visual Component Library (VCL) from previous versions of Delphi is available, as well as new and enhanced VCL components.

The Delphi language for Win32 now includes many new features that previously were in only the .NET version of Delphi. For example, function inlining can significantly boost performance. Delphi 2005 also includes the “classic mode” floating designers popular in previous versions of Delphi.

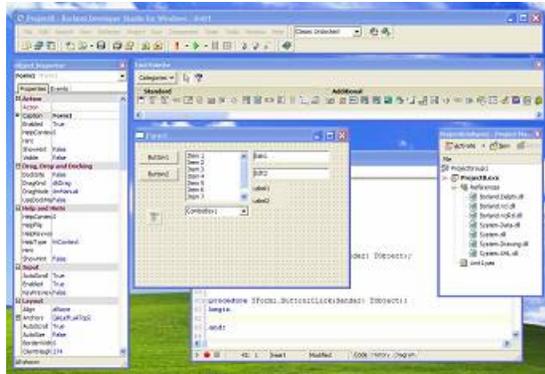


Figure 2: ‘Classic Mode’ floating VCL designers for Win32 applications

C# or Delphi for the .NET Framework

The future of Windows development is .NET, and Delphi 2005 provides outstanding support for the full .NET Framework SDK. Delphi includes full graphical designers for ASP.NET, Windows Forms, Web Services, and component creation, using either the Delphi or C# languages.

The .NET Framework personality of Delphi supports both the .NET SDK (Windows Forms, Web forms) and the Borland Visual Component Library, VCL.NET. Porting VCL applications from Win32 to the VCL.NET library is quite easy.

In addition to VCL components, Delphi 2005 includes .NET versions of all the popular Delphi VCL data access technologies to make moving database applications from Win32 to .NET easier than any other solution. For example, existing Delphi Win32 applications that use Borland® dbGo™ for ADO can now easily port to .NET with the new dbGo.NET components.

```
1 unit Unit1;  
2 interface  
3 uses  
4   Borland.Vcl.*;  
5 type  
6 TForm1 = class(TForm)  
7 private  
8   / Private declarations /
```

Figure 3: New Delphi for .NET language features include multi-unit namespaces



Figure 4: The Project Upgrade wizard for VCL applications upgrades to .NET or Win32

This combination of personalities means that project managers can take better advantage of the skills of their team within a single IDE.

```
7 var  
8   i: String = 'hello world';  
9   c: Char;  
10 begin  
11   for c in i do  
12     write(c);  
13 end.
```

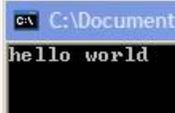


Figure 5: New Delphi for Win32 language enhancements include for.. in

Delphi: The language evolves and grows

Delphi 2005 now includes new language features that are available to both Win32 and .NET developers. For instance, the for .. in construct iterates through enumerated types quickly and simply.

Streamlined environment for fast code development

The Delphi 2005 IDE is based on a deep history and understanding of how developers work. Many touches in the IDE make it possible to get the job done quickly, either by visually using the graphical designers or streamlining the process of writing code.

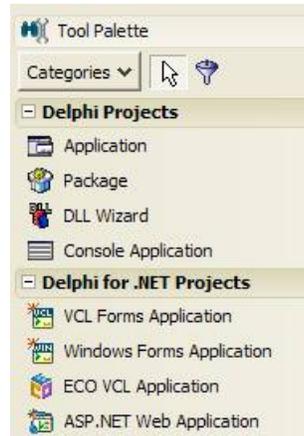


Figure 6: *The new Tool Palette makes a range of project wizards one mouse click away*

For example, the new streamlined Tool Palette shows components from the .NET Framework or the Delphi VCL together with wizards that create new forms or source code units. All these are either a mouse click or keystroke away, eliminating the need to use the mouse to add components or run application wizards.

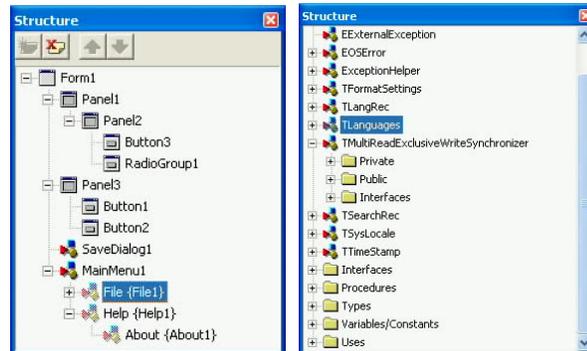


Figure 7: The Structure Pane shows the relationship between objects or components

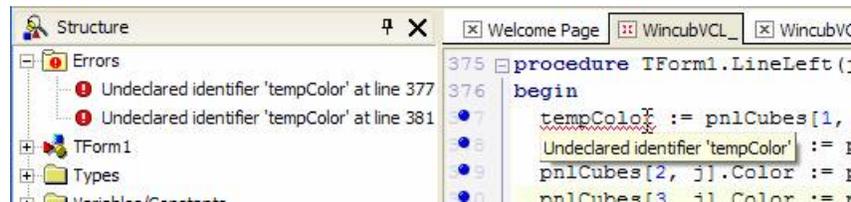


Figure 8: The Structure Pane and tool-tip error messages speed coding

Sophisticated new features in the IDE such as dynamic ErrorInsight,™ tool-tip error messages for source code errors, and a detailed listing of all errors in the ErrorInsight window help avoid syntax and semantic issues as you type and before compilation. During debugging, a color-coded event log, support for multiple stack frames, and disassembly of .NET Framework IL help quickly pinpoint difficult-to-find bugs.

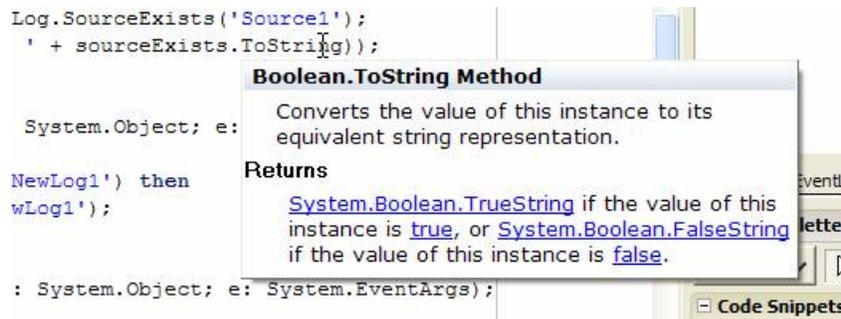


Figure 9: In-line tool tips display quick help on methods

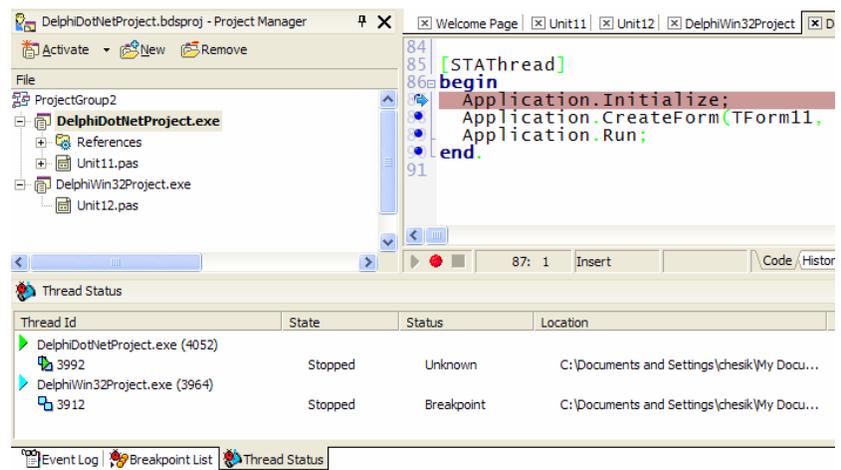


Figure 10: Simultaneously debug Win32 and .NET programs

The multiple personalities of Delphi 2005 extend to the debugger. You may even debug project groups with both Win32 and .NET systems simultaneously. Many debugger enhancements make it possible to trigger breakpoints when data changes to a particular value output a call stack to the trace log. Delphi 2005 makes it possible to attach to running processes and debug those in place.

Refactoring and beyond: Outstanding code productivity

A key requirement for teams that must respond rapidly to requests for change is the ability to create small, incremental releases of software. Agile programming techniques make this possible by encouraging developers to gradually refine the system iteratively.

Because these small iterations are delivered to users frequently, the code must be correct at each point, and developers in a team must communicate effectively as the changes are made.

Delphi 2005 includes innovative features for Delphi and C# to assist in refactoring program code. At each stage, the refactorings ensure that the code works the same before the program was changed.

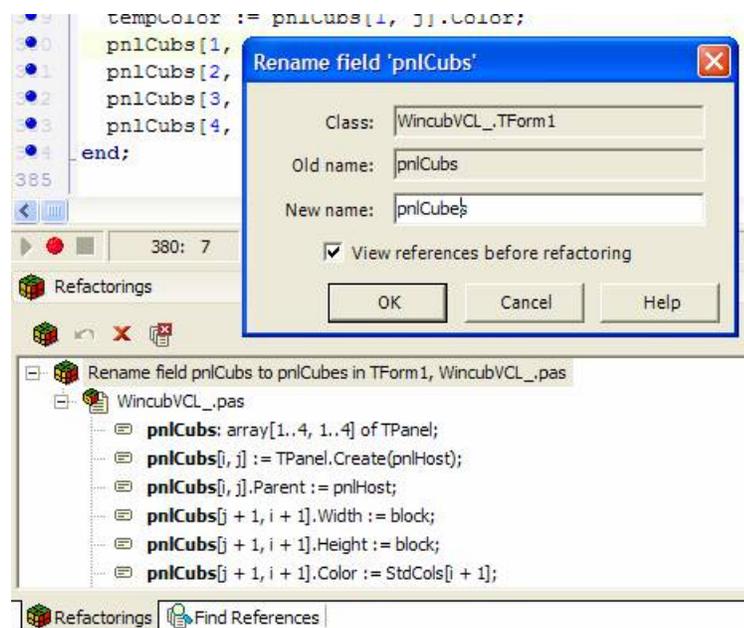
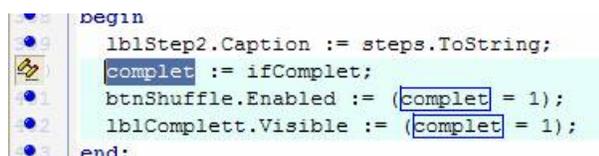


Figure 11: *Easy, fast, accurate refactoring includes global rename of symbols*

For small changes to blocks of code, SyncEdit makes it much easier to rename an object or member. The developer marks the block of code and changes one copy of the name. Delphi

2005 automatically changes the name everywhere else in the block. Over a typical project's duration, this feature alone can save hours of typing.



```
begin
  lblStep2.Caption := steps.ToString;
  complet := ifComplett;
  btnShuffle.Enabled := (complet = 1);
  lblComplett.Visible := (complet = 1);
end;
```

Figure 12: SyncEdit automatically changes several references to a single object or member

Wider changes are covered by a range of refactorings—for example, extracting methods, extracting strings and turning them into resources, global renames, and automatic method or variable declaration. In each case, a preview shows the developer exactly what will be done by the refactoring.

Figure 13, for example, shows the Extract Method wizard, which takes a block of in-line code and turns it into a separate method. The wizard automatically determines the parameters and local members for the new method and inserts a call to the method back in the original block of code.

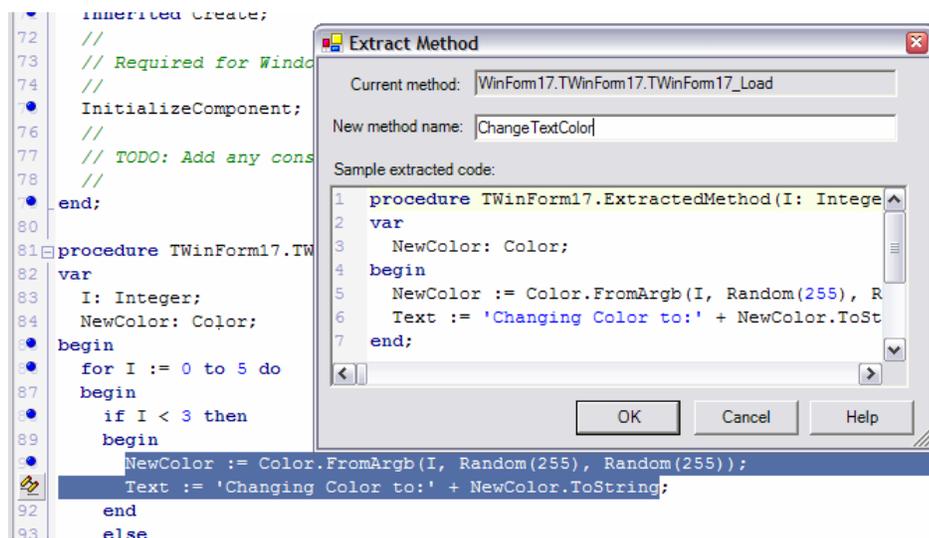


Figure 13: Refactoring improves code structure—for example, extracting methods

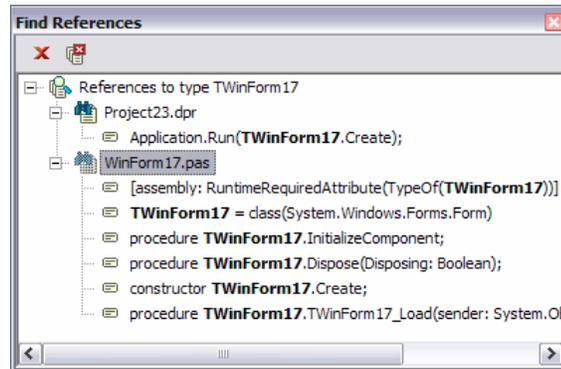


Figure 14: *Locate code references fast to help debugging and code reuse*

In Delphi 2005, not only are the refactorings fast and easy to use, they also ensure that what the code developers create is correct.

Guaranteed code quality with integrated unit tests

When refactoring is a daily part of agile programming, developers must ensure that code still works as it should. Automated unit tests go hand-in-hand with the rapid code iterations that agile programming demands.

A unit test checks a class to make sure that it behaves as the developer expects. A few tests are run against the class to ensure that, for example, range checking is properly handled, exceptions are raised in the expected way, and any calculations are correctly performed.

Delphi 2005 includes both the popular NUnit unit testing framework for C# and Delphi .NET Framework unit testing and the DUnit framework for Delphi for Win32. Comprehensive wizards simplify creation of the test project and test cases within that project, making it easier for developers to use this important discipline.

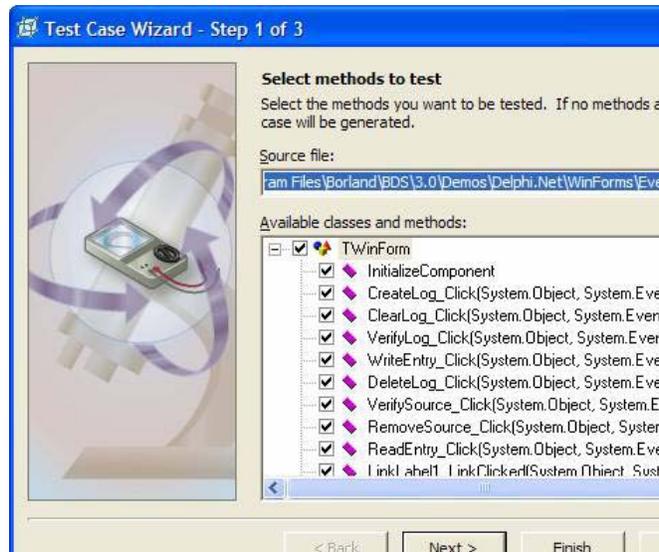


Figure 15: Create unit tests automatically with the new Test Case Wizard

With a good set of unit tests, you can quickly verify that changes to a class don't break anything else in the system. Frequent unit testing is valuable for individual programmers, but it's when individuals combine to form a team that they can really help everyone pull together.

Performance, baked in from the start

Quality code doesn't just run correctly; it runs quickly. Delphi 2005 Architect edition includes Borland® Optimizait™ Profiler for the .NET Framework, so developers can include application profiling as part of their testing procedure.

When profiling is integrated closely with development, developers can make sure that their initial application design is correct and efficient. It's much easier to profile while the application is constructed than to attempt to fix performance issues later in the process.

Check out, refactor, unit test, check in:

Team development, simplified

With the unit testing and refactoring tools in Delphi 2005, several developers can work on quite small pieces of the application. This demands effective, efficient communication among team members.

At its heart, Borland® StarTeam® has a powerful “merge” capability that enables developers to work together on the same units of source code without locking large parts of the repository. StarTeam lets many developers work on the same part of the project and automatically resolve conflicts at check-in.

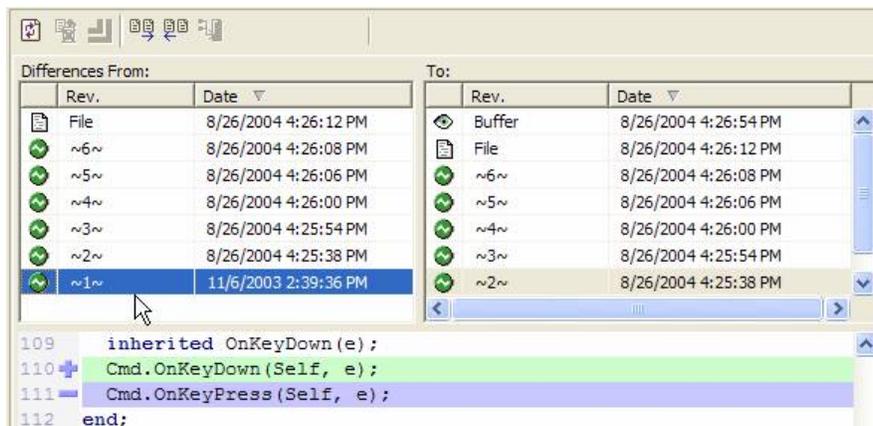


Figure 16: Keep track of code changes with the History tab for each source code unit

Delphi 2005 includes a “History” tab for each source file to keep track of changes, you can roll back changes and see differences between them. StarTeam also allows individual change requests and bug reports to be associated with each file, greatly streamlining team coordination.

When it’s time to release a project, StarTeam has comprehensive features for tracking build labels and, if necessary, multiple branches of a system.

Delphi includes an innovative safety feature: automatic local versioning for maintaining source change history. To protect you from accidental loss, Delphi 2005 maintains a local history of changes made to source files. You can back out intentional or unintentional modifications, back out refactorings, or simply revert to a particular date. At any time, you can conveniently inspect the history of source code, the differences between the current version and the historical version, and roll back to any previous version.

The History Manager supports both automatic local backups and StarTeam check-ins.

Better technology for faster development

Delphi 2005 is more than the most efficient development environment for Windows. It also includes a broad range of technologies that simplify applications. This allows developers to focus on the particular business needs of an application without worrying about underlying infrastructure.

Such a broad range of technologies makes developing Windows applications much faster—with much higher quality.

Delphi RAD for ADO.NET

Virtually every enterprise application needs to access a database, and Delphi RAD for ADO.NET greatly extends the ease of use and capabilities of the Microsoft ADO.NET framework for Delphi and C# developers.

The cornerstone of Delphi RAD for ADO.NET is Borland® Data Provider (BDP). Because the ADO.NET drivers provided by Microsoft, Oracle, and others use their own data types and bindings, changing database types is a major undertaking. Code must be rewritten for each database. For example, the Microsoft® SQL Server™ ADO.NET driver provides a `SqlDbType` class for data, while the Oracle driver uses a `OracleData` class. Changing from one database to another requires that all of these object types change, too.

Borland Data Provider in Delphi 2005 eliminates this difficulty. The BDP ADO.NET driver is generic, so database code must be written and tested only once. A different database back end can be used simply by changing the BDP connection string.

By using Borland Data Provider, an application can be shrink-wrapped and deployed to different database types, which might be in different departments. A central office might use Oracle, for instance, while a branch office uses SQL Server Developer Edition. Or a team might develop against one database and deploy using another. ISVs and VARs instantly broaden the appeal of a product written with the BDP, because it can target the leading databases in use in enterprises today without rewrites.

BDP in Delphi 2005 supports Microsoft SQL Server 2000 Enterprise and Developer editions, Oracle® 10g, Borland® InterBase,® IBM® DB2,® and Sybase.® Because it's an open specification, third-party drivers can also be created for other databases.

The integrated BDP database explorer includes many powerful features, including schema creation, store procedure testing, and support for migrating tables and data between databases—even among different vendors. This ability to work with metadata is database-independent, so developers can create and change database structure regardless of the actual server type.

Distributed data with DataSync and DataHub resolver and provider

Whether using BDP or vendor-supplied ADO.NET providers, a developer can take advantage of the powerful distributed RAD data capabilities included with Delphi 2005. These make it possible to easily create scalable multi-tier applications with a disconnected database architecture using .NET remoting and ADO.NET.

The DataSync and DataHub components sit between the user interface (either Windows Forms or ASP.NET) and back-end databases, handling conflict resolution and exceptions as necessary.

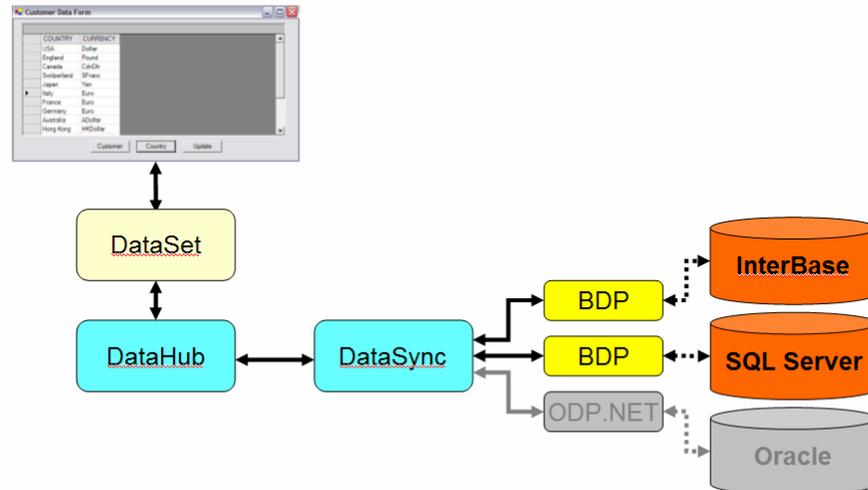


Figure 17: DataHub and DataSync components map multiple enterprise databases into a single remotely connected DataSet

The RemoteServer and RemoteConnection components can sit between the DataHub and DataSync provider, a completely component-based solution for multi-tier ADO.NET development. The RemoteServer publishes DataSync components in one process to applications that use RemoteConnection components connected to DataHubs. This allows a true distributed database architecture to integrate with any ADO.NET data source.

The Data Migration Wizard

Enterprises typically have a mix of many databases that must be combined, creating a major challenge for systems that must work with existing data. To simplify this task, Delphi 2005 includes the new Data Migration Wizard and Data Explorer.

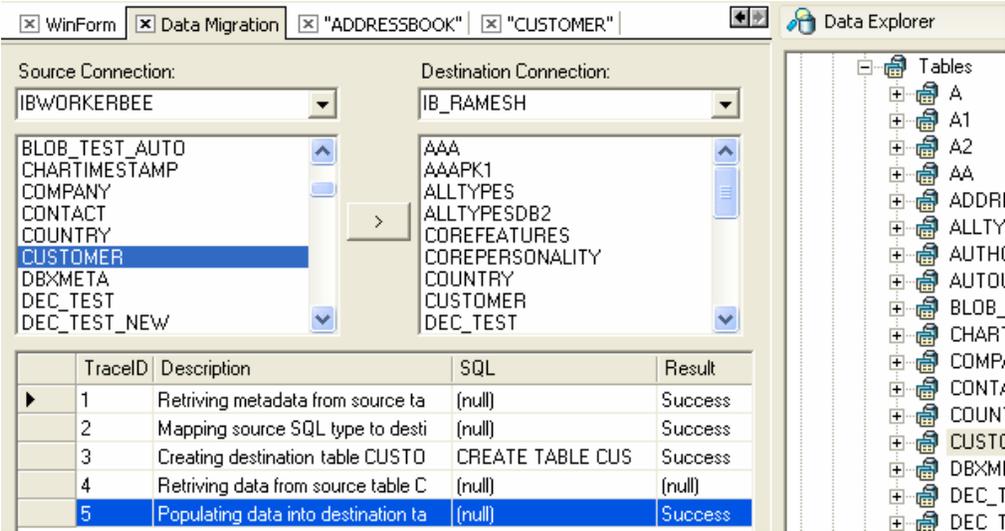


Figure 18: Simplifying data migration

Together, these provide a mechanism for developers to work with multiple databases and translate data from one point to another. This is database-independent and can work with any ADO.NET data source.

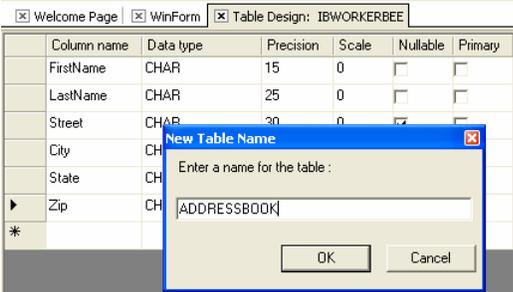


Figure 19: Create and modify any ADO.NET data source with the new Data Explorer

The Data Explorer includes new capabilities to create and maintain database metadata, so developers can easily and quickly maintain database structures from within the Delphi 2005 environment. It also includes capabilities for testing stored procedures, invaluable when optimizing applications for peak performance. Because the Data Explorer works with the BDP, the same metadata and stored procedure tools are available for the full range of databases supported by the BDP, greatly simplifying maintenance.

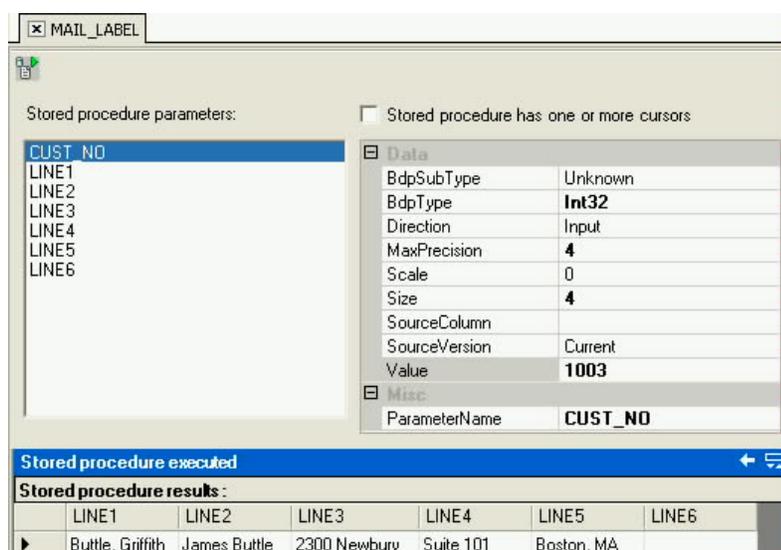


Figure 20: Execute and debug any ADO.NET stored procedure

Enhanced and upgraded Win32 database support, and easy migration to .NET

Delphi has always had powerful data access capabilities. Over the releases of Delphi for Win32 these capabilities have been carefully tuned to deliver optimum performance for a range of database technologies, from the Borland® Database Engine in Delphi 1 through dbExpress™ and InterBase Express™ in later versions.

Each of these technologies is available for the Win32 and .NET Framework personalities of Delphi 2005. When using the VCL.NET framework, all of the Delphi database technologies

are now available, making it easy for developers to move even complex business applications to .NET. In many cases, only a recompile is necessary.

This support for VCL database technology in Delphi 2005 now includes Borland® dbGo.™ This powerful driver, introduced for Delphi 7, gives high-speed access to Microsoft ADO from VCL components. dbGo.NET brings superb direct ADO capabilities to the .NET Framework for the first time.

Simple development and deployment of Web applications

Many systems now use Web pages for some or all of their user interfaces. Delphi 2005 includes powerful capabilities to build Web-based applications for ASP.NET or for Win32.

Streamlined Web forms for ASP.NET

For designing ASP.NET applications, Borland has enhanced Web Forms by including a set of controls, called dbWeb, which can work with database data directly. Live data is shown during design time, which simplifies development of Web applications. These controls automatically handle aggregation in data grids and cascaded updates and deletes across multi-table forms, greatly simplifying application logic for data-driven ASP.NET applications.

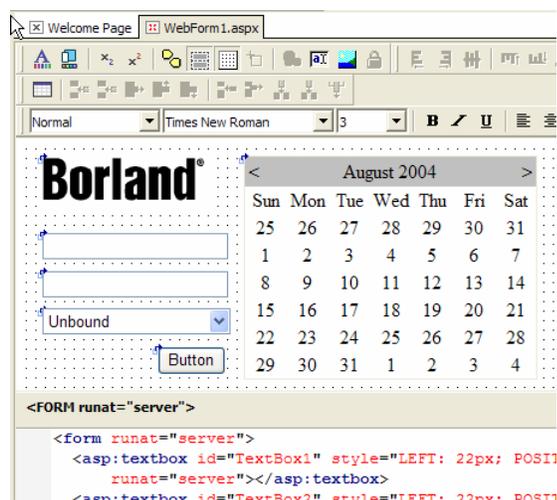


Figure 21: Easy editing of ASP.NET applications

For increased flexibility, Delphi 2005 includes new wizards to assist in the creation of new custom dbWeb ASP.NET controls. A new Navigation API eliminates much custom navigation code.

Atozed IntraWeb: Fast RAD Web form design for Win32 and .NET

Delphi 2005 also includes a powerful technology for Web browser applications that use the Delphi VCL. IntraWeb, available for Win32 and .NET, gives developers an easy-to-use drag-and-drop forms-based development environment for HTML form design. These forms can then be run directly from either an IIS or an Apache Web server.

Because IntraWeb includes its own server-side components, you can take considerable control of the low-level aspects of a Web program. IntraWeb automatically takes care of details such as saving state on the server during a session, simplifying code and allowing developers to concentrate on the application rather than on systems plumbing.

Deploy Web applications easily, quickly, and accurately

A challenge often faced by Web developers is deploying a completed application. Frequently, a large number of files comprise a Web-based system, and they might have to be deployed to a local file share or, more typically, a remote computer using FTP.

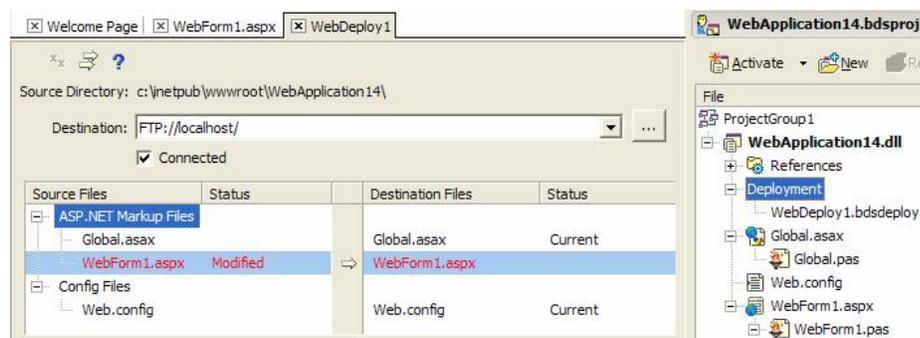


Figure 22: Simple and easy deployment of Web applications

Delphi 2005 includes a new Web Deployment Manager that can automatically deploy a Web application and its associated assets to any Web server, through a regular Microsoft® Xcopy, a Microsoft® FrontPage share, or through FTP. This wizard works for .NET Framework ASP.NET applications and for Delphi IntraWeb applications.

Beyond databases: Model-powered development with Borland® Enterprise Core Objects (ECO™) II

Traditional client-server application development works directly with relational databases and creates user interfaces that add, edit, and remove records from related tables inside these databases. Like other application development tools, Delphi 2005 includes many features that make the process of development using databases as productive as possible.

However, for ultimate productivity, Delphi 2005 provides Borland® Enterprise Core Objects (ECO™) II, the new way to build applications that is especially well-suited to business applications.

ECO is completely model powered. Rather than creating database tables, an architect uses a graphical UML designer to create diagrams describing the classes in the application and their relationships. These classes are then available as .NET objects for C# or Delphi for .NET developers to use in their code. Relationships among the objects, such as associations, inheritance, and attributes, are defined in the model and make implementing business logic easy. Following associations from one class to another becomes as simple as using an attribute without the need to use keys.

The ECO framework also offers services to manage the persistence of these objects, storing them in any ADO.NET-connected relational database, or, if preferred, in XML files. The object-relational database mapping can, if desired, be completely transparent. Effectively, the developer merely works with objects and relies on the ECO framework to store or retrieve them as needed.

Some situations require more detailed control over the object-relational mapping. The model might have to map onto an existing database. ECO II can set up mapping schemas containing

all of the information needed for the ECO framework to perform the object-relational mapping in runtime. These schemas can be automatically created from a design model, or they can be reverse-engineered from an existing database together with the required source code and model. As a result, ECO II makes it easy to add object-relational capabilities to an existing database.

Once the model has been constructed using UML or the reverse-engineering capability, the developer can directly bind Windows Forms or ASP.NET controls to the ECO objects.

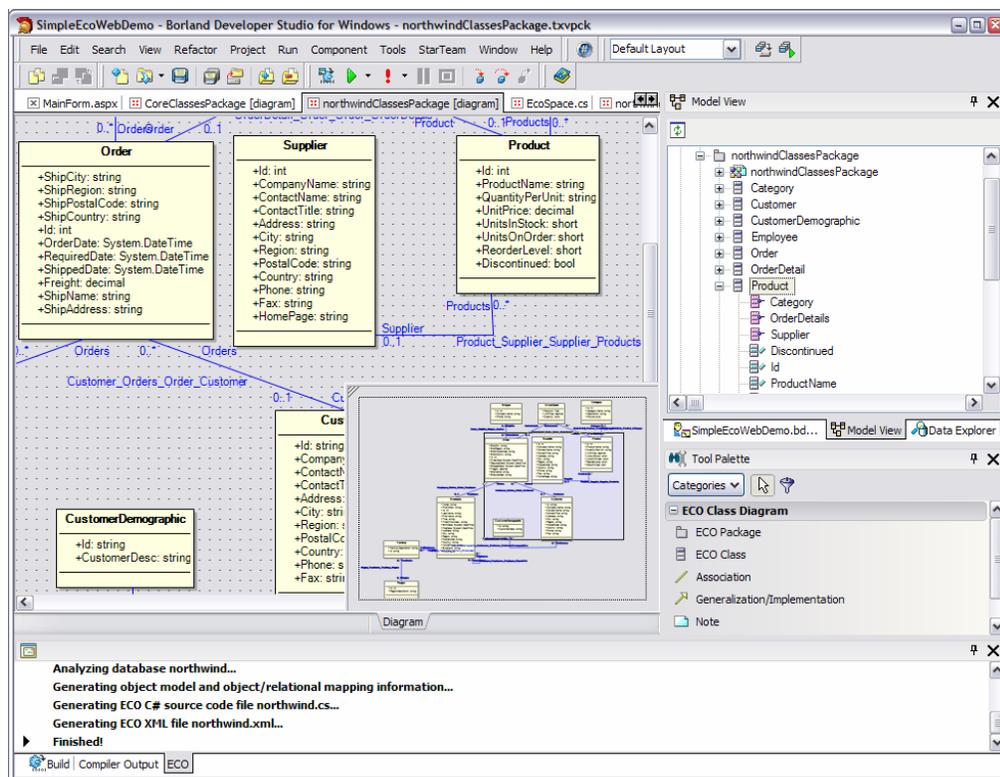


Figure 23: The Microsoft SQL Server Northwind database reversed to a UML class diagram, ready to power an ECO II application

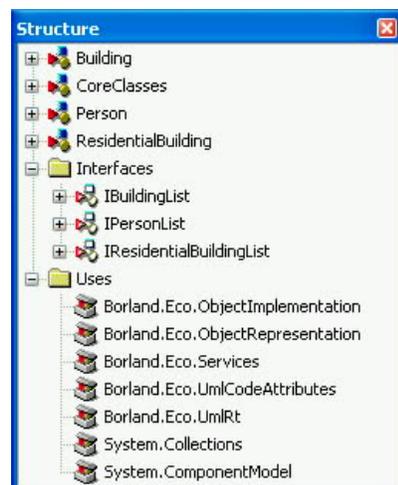
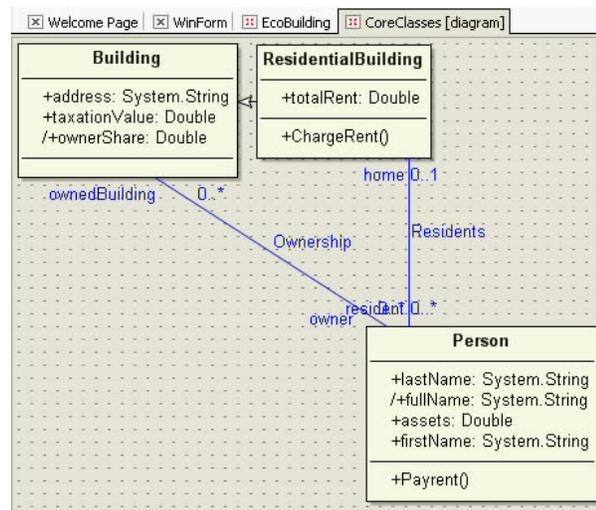


Figure 24: ECO II uses UML models to directly create and maintain .NET classes

While the ECO framework offers several common services required by most applications, such as persistence, caching, inter-object relationship management, subscriptions, and transactions, it also includes more advanced services such as undo/redo and object versioning. All of these services are performed when required and configured from the model, making the system easy to maintain.

Choosing ECO for an application can dramatically increase application development speed. No longer must developers spend time creating and maintaining relational database schemas, maintaining class and object relationships, and translating architecture diagrams to real-world implementations in source code. ECO will even evolve database structures as the model evolves, saving significant development time.

In Delphi 2005, ECO supports both ASP.NET and Windows Forms applications. The persistence manager is multi-threaded, so it can support highly scalable, high-availability applications.

Indeed, multiple ECO ASP.NET servers can be used to share a single persistence mapper through the new Persistence Mapper Synchronizer for ultimate performance. The application architect is given greater flexibility to control memory management and session state maintenance, allowing optimization of performance and concurrency.

Beyond Windows: Integration with other systems

As we've seen, increased enterprise development means that more applications are being created, which means that connecting to existing systems is becoming increasingly more important.

Many development teams have a need to link Microsoft® .NET Framework applications to Java™ systems. Delphi 2005 includes a range of technologies that make creating this link straightforward.

For loose integration between systems, Web Services can be a practical option. One system publishes a list of services that it can provide, and the other side consumes these services, working with the remote objects directly.

Delphi 2005 can both create and consume Web Services and includes many features to do so. These include a WSDL importer to directly create Delphi or C# code to use a Web Service and wizards that create Web Service server and client applications for this WSDL.

More direct integration is possible through Borland® Janeva.™ In Delphi 2005, wizards can directly inspect an Enterprise JavaBean™ (EJB™) and discover classes inside it. These can then be used directly from Delphi 2005 as if they were native .NET Framework classes. Janeva handles the marshaling and mapping between the EJB and .NET Framework.

Delphi 2005 also supports CORBA® through a Delphi-language IDL mapping in the included Borland® VisiBroker® for Delphi product. This allows both client-side consumption and server-side provision of CORBA objects directly from Delphi components.

Delphi 2005: The broadest solution for Windows development

Ten years in the making, Delphi 2005 represents the pinnacle of Windows development technology from Borland.

Historically, Delphi has often been used to good advantage in production systems where human input is necessary. This reflects the excellent capabilities in Delphi to optimize user interfaces, both rich-client and Web-based, and the further ability to easily define and work with business rules. Delphi is therefore a strong choice for any team that needs a fast, easy-to-use, easy-to-customize solution for Windows development.

Delphi also has powerful server-side capabilities. The low-level control given by many of the components and the powerful integration to Java, CORBA, and Web Services makes Delphi especially valuable for business-object programming and complex middleware. Here, too, the ability to create .NET Framework or Win32 applications gives Delphi teams a great advantage.

Delphi 2005 includes many features for .NET development, regarded as the future of Windows. It now supports C# and Delphi for .NET languages, giving the broadest range of choice for developers who need to use a range of SDKs and languages.

Delphi 2005 also supports and greatly enhances Delphi for Win32 programming. Borland, which recognizes that many Win32 projects will be developed and maintained in the coming years, will continue to add new features to Delphi for Win32. As demonstrated with Delphi 2005, such updates promise to include new language features, productivity features, and interoperability between Win32 and .NET systems.

The technologies in Delphi extend and improve each part of the .NET Framework. Whether Delphi is used for Windows Forms programming, ASP.NET Internet systems, Web Services, or component writing, Delphi includes a set of technologies to streamline and speed development.

With the resurgence in demand for internally developed systems, the ability to quickly adapt to change is more important than ever. The support in Delphi 2005 for easy refactoring, integrated unit tests, efficient code profiling, and proper source code management makes it easy for teams to adopt agile programming methodologies. With Delphi 2005, teams can write better code and respond to user requests faster than with any other Windows programming environment.

Good enterprise systems require quality code from a development team that is agile and efficient. For Windows, Delphi 2005 helps teams achieve this goal: it is the most comprehensive and productive development environment available.

Made in Borland® Copyright © 2004 Borland Software Corporation. All rights reserved. All Borland brand and product names are trademarks or registered trademarks of Borland Software Corporation in the United States and other countries. Java and all Java-based marks are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries. Microsoft, Windows, and other Microsoft product names are trademarks or registered trademarks of Microsoft Corporation in the U.S. and other countries. All other marks are the property of their respective owners. Corporate Headquarters: 100 Enterprise Way, Scotts Valley, CA 95066-3249 • 831-431-1000 • www.borland.com • Offices in: Australia, Brazil, Canada, China, Czech Republic, Finland, France, Germany, Hong Kong, Hungary, India, Ireland, Italy, Japan, Korea, Mexico, the Netherlands, New Zealand, Russia, Singapore, Spain, Sweden, Taiwan, the United Kingdom, and the United States. • 22951