EMBARCADERO TECHNOLOGIES®

RAD Studio 2010

Product Review Guide

August 2009

Corporate Headquarters
100 California Street, 12th Floor
San Francisco, California 94111

EMEA Headquarters
York House
18 York Road
Maidenhead, Berkshire
SL6 1SF, United Kingdom

Asia-Pacific Headquarters
L7. 313 La Trobe Street
Melbourne VIC 3000
Australia

TABLE OF CONTENTS

INTRODUCTION

Thank you for taking the time to review Embarcadero RAD Studio 2010. This guide is designed to inform you about the capabilities, features, and attributes of the product so that you can be fully informed when writing your review. The guide is divided into three parts. The first gives a general overview of RAD Studio 2010, describing its basic purpose and capabilities. This section will be a more high-level overview of RAD Studio 2010's features. The second part is a summary of the major new features in RAD Studio 2010. This part will discuss the things that are the "latest and greatest" in RAD Studio 2010 – i.e. the compelling reasons for buying an upgrade. The third section will give a more complete description of a selection of the product's extensive feature set.

GENERAL OVERVIEW OF RAD STUDIO 2010

Embarcadero RAD Studio 2010 is a general purpose, rapid application development (RAD) Windows application development tool. The RAD Studio package is made up of 3 different products; Delphi® 2010, C++Builder® 2010, and Delphi Prism™ 2010. RAD Studio's Delphi and C++Builder tools produce native Win32 binaries that execute on x86 operating systems. Delphi Prism produces .NET and cross-platform Mono applications. RAD Studio 2010 supports the Delphi language (Object Pascal), C++, and the Delphi Prism language (aka Oxygene), an Object Pascal based language for .NET.

With RAD Studio 2010, developers can build almost any type of Windows binary, including stand-alone executables (EXEs) and dynamic link libraries (DLLs), OCX and COM objects, type libraries, Control Panel applets, Windows Service applications, and console applications, and including full .NET support. Developers can build client applications with rich, complicated user interfaces, or simple command line applications. They can build database client applications that speak directly to major relational database management systems (RDBMSs), and middle-tier application servers, web applications, web sites and web services, windowed applications, ActiveX controls, and multi-threaded applications for running complex embedded systems. In short, RAD Studio 2010 can meet the needs of any developer writing any application for Windows. It takes a long time and some very serious effort to reach any "virtual programming walls" developing with Embarcadero RAD Studio 2010.

WHAT IS NEW IN RAD STUDIO 2010

Radically reduce development time

Every feature in the RAD Studio development environment is designed to speed coding so you can complete your projects faster. RAD Studio 2010 will make you even more productive and save you time with enhancements throughout the IDE:

- IDE Insight for easy access to all IDE features, settings and components without searching through menus and dialogs
- Code Formatter to enable consistent coding styles with less work
- C++ Class Explorer for quick navigation and management of classes in your project
- Improved Search and File Reopen to quickly find the information you need
- Debugger data visualizers make debugging easier by customizing the display of data types in the debugger

- Debugger thread control for freezing, thawing and isolating threads as well as setting breakpoints for selected threads so you can track down problems
- New Debugger Options: "Scroll new events into view" and "Ignore non-user breakpoints"
- New introductory code audits and metrics in Delphi Professional for better understanding code and project health; full set of audits and metrics in Enterprise and Architect

Free your customers from keyboards

Rapidly build touch based GUI, tablet, touchpad, and kiosk applications or easily upgrade existing applications UIs with little or no additional coding.

- Pluggable gesture engine architecture
- Works on all supported versions of Windows (2000, XP, Vista and Windows 7)
- Use touch-enabled hardware or work with what you have (e.g. mouse)
- Integrated support for touch and multi-touch interfaces in the base VCL
- 30+ standard gestures for panning, zooming, rotating and more
- Create your own with the Custom Gesture Editor
- Touch Keyboard - a complete virtual keyboard for enhanced non-keyboard interface interactions that supports multiple locales and languages

Make the connection with data and apps

With RAD Studio 2010 you'll make the connection with more data, more Web Services, and more application architectures

- New Firebird 2.1 and 1.5 support in dbExpress
- Updated drivers for InterBase 2009, Microsoft SQL Server 2008, Oracle 11g and MySQL 5.1
- Take advantage of Web Services functionality from Amazon and others with new SOAP 1.2 client support
- HTTP communication and in-process connectivity for DataSnap multi-tier applications
- New DataSnap wizards from the Object Gallery to make server creation even easier
- DataSnap callbacks enable servers to communicate with clients
- Filtering allows for complete control over the data stream between clients and DataSnap servers
- Participate in popular application architectures with REST and JSON values support in DataSnap

Reach more user desktops

Support more Windows desktops without worrying about the specific Windows API details of each version and support users worldwide with Unicode throughout the development environment and improved language support.

- VCL controls are optimized to take advantage of the capabilities and theming of XP, Vista and Windows 7
- Fully Unicode enabled throughout to handle worldwide data and users
- Deliver touch and mouse gesturing enabled apps on Windows XP and Vista and take advantage of the new touch support in Windows 7
- Expanded Open Tools API for building plug-ins to the Delphi IDE
- English, German, French and Japanese translations available for IDE menus and dialogs, compiled units, resources, and source code
- Easy switching between languages for IDE menus and dialogs and more

Code and compile like never before

Delphi 2010 includes new RTTI support and new compiler and language enhancements

- RTTI support for exposure of Methods, Fields, and Properties to support dynamic invocations and other meta-programming approaches
- Object-oriented file and directory IO classes
- Custom attribute support for most code elements - types, fields, properties, methods and parameters
- Enhanced TStringBuilder for easier and faster string concatenation and manipulation
- Enhanced generics with full RTL list and collection support
- Enhanced support for localized resources
- Background compilation so you can continue working while you compile

C++Builder 2010 further builds on previously introduced C++0x language features with even more compiler and library enhancements:

- FastMM is now the standard heap manager for C++ runtime libraries
- Support for #pragma once
- -Zx option for generating XML representation of source code
- Added support for _FUNCTION_
- Support for [[deprecated]] attribute
- __declspec(dllimport) and __declspec(dllexport) for template classes
- Update Boost libraries 1.39
- Improved standard C++ heap manager
- Optimized string/memory handling functions
- Background compilation so you can continue working while you compile

More data modeling power in RAD Studio 2010 Architect

RAD Studio 2010 Architect edition includes new ER/Studio 8.0 Developer Edition to help users discover, document, and re-use data assets and gives you the power to easily reverse-engineer, analyze, and optimize databases. New ER/Studio features include:

- Visual Data Lineage - Visually analyze and document how data flows through your organization without needing to inspect code
- Attribute-level Submodeling - Choose which attributes/columns to include in the entities/tables in the submodel and also describe submodels on the new Definition tab, then create queries to search on the definitions
- Object-level Compare Flags - Indicate intentional discrepancies when comparing models that the Compare Wizard should ignore
- Produce reports in HTML format
- Microsoft SQL Server 2008 support

These new features will be covered more fully below.

New for .NET development (Delphi Prism)

Go further with the Delphi Prism language

The Delphi Prism language is a great way for Delphi developers and .NET developers to write .NET applications. Delphi developers can take advantage of familiar syntax and .NET developers will find exciting language features not available in other .NET programming languages. The Delphi Prism language has been updated with the following new features:

- Enhanced compatibility with the Delphi language

- Aspect Oriented Programming (AOP)
- Support for Dynamic Typing (under .NET 4.0)

Major Compiler Features

- RemObjects Cirrus: AOP for Oxygene
- Standard Aspects Library for RemObjects Cirrus (shipping as Tech Preview)
- Unmanaged Exports
- Generic Type Variance
- Volatile fields
- CLSCompliantAttribute support and compiler warnings
- New LINQ Query Expressions operators Skip, While, Take and Take While

Minor Compiler Features

- Unquote expression support (mainly to support Cirrus)
- $DELPHICOMPATIBILITY compiler directive and project option
- Runtime range checking
- Range Enum Types (e.g. "type Ten = 1..10;")
- Read-only classes
- Support for negative low bounds in Array Types

Other New Features

- CodeDom enhancements to support OxygeneInterface, OxygenePartial and OxygeneEmpty tags in UserData
- Pre/Post build events
- Debug options: Remote Machine
- Custom "Add Reference" pane for adding Mono assembly references
- New Monobjc Template and Monobjc libraries deployed with Delphi Prism
- Updated setup to install the latest Mono 2.4
- Added Internet Pack as an integrated part of setup
- Integrated F1 help based on an offline copy of the documentation wiki
- Added templates for ASP.NET Web projects

Make the connection with data and applications

With Delphi Prism 2010, developers can build .NET DataSnap® client applications that use new capabilities of DataSnap 2010, including support for communicating with the server using the HTTP protocol. DataSnap features are available in Delphi Prism Enterprise, Embarcadero RAD Studio Enterprise and Embarcadero RAD Studio Architect.

Features vary by product edition. See the RAD Studio 2010 Feature Matrix for a full list of features by edition.

A WORD ON DELPHI PRISM

The Delphi Prism product is not covered in this reviewers guide but will be covered in a separate guide as it is focused 100% on .NET. The underlying IDE is based on Microsoft's Visual Studio Shell (VSS) program that is similar to the Eclipse IDE framework. Since Delphi Prism uses a different interface and approach, it is believed that a separate document would be better for that part of RAD Studio. In the future it may be added to the end of the document. This document will focus on both Delphi and C++Builder products as they are both built on the same internal IDE framework and are focused on native applications.

PREREQUISITES

To install RAD Studio 2010, the following prerequisites must be installed:

- Microsoft .NET Framework 2.0 or later
- Microsoft Direct Access Objects 2.8
- Microsoft Internet Explorer v6.0 SP1 or later
- Microsoft XML Core Services (MSXML) v4.0 SP2 or later
- Microsoft Visual J# .NET v2.0 Redistributable

If you don't already have the prerequisites installed on your system, the Delphi installer will install them for you.

The .NET Framework is required by the IDE, but note that native applications built with Delphi has no dependencies at all on the .NET Framework.

MINIMUM SYSTEM REQUIREMENTS

The following system requirements are recommended for running RAD Studio 2010:

- Intel Pentium or compatible, 1.4Ghz minimum (2Ghz+ recommended)
- 1GB RAM (2GB+ Recommended)
- 3GB free hard disk space for Delphi
- 750MB free hard disk space for prerequisites
- DVD-ROM Drive
- 1024x768 or higher resolution monitor
- Mouse or other pointing device

The following Windows platforms are supported for installing RAD Studio 2010:

- Microsoft Windows XP Home or Professional (SP3 or higher)
- Microsoft Windows Vista SP1 (requires Administrator rights)
- Microsoft Windows Server 2003 (SP1)
- Microsoft Windows Server 2008
- Microsoft Windows 7 (requires Administrator rights)

Note: The English version of RAD Studio 2010 updates Internet Explorer versions earlier than 6.0 with the English version of Internet Explorer 6.0 SP1. If you are running a localized operating system, run Windows Update to ensure that you get the proper localized version of Internet Explorer.

INTERNATIONALIZATIONS

The RAD Studio 2010 product is fully localized for the following languages.

- English
- French
- German
- Japanese

EDITIONS

RAD Studio 2010 has three different editions to cater to different market segments – Professional, Enterprise, and Architect.

PROFESSIONAL

The Professional edition of RAD Studio 2010 is designed for developers who need a general purpose, RAD development tool with limited or local-only database access. The Professional edition includes the full-featured IDE, both the Delphi and C++ languages, and the complete VCL, including the VCL source code and integrated Unit Testing. It includes local-only database access to InterBase, MySQL and Blackfish SQL.

The Professional edition also contains a limited version of VCL for the Web -- applications are limited to five connections and stand-alone servers. The Professional edition also includes limited functionality that supports Unified Modeling Language (UML) for Reverse engineering of class diagrams, and static code analysis tools for software metrics and audits.

The Professional edition is attractive to Independent Software Vendors (ISVs), professional developers without a need for remote database access, and any developer wanting to develop general Windows applications and utilities.

ENTERPRISE

The Enterprise edition of RAD Studio 2010 is designed for developers who need access to enterprise level data stored in RDBMSs and includes all features from the Professional edition. The Enterprise edition provides native local and remote access to nine different database engines:

- Firebird 2.1 and 1.5 support
- InterBase 7.5.1, 2007 and 2009
- Blackfish SQL for .NET andJava
- Oracle 10g and 11g
- Microsoft SQL Server 2000, 2005 and 2008
- DB2 UDB 8.X
- MySQL 4.0.x and 5.1.x
- Informix 9x
- Sybase Adaptive Server Enterprise 12.5
- Sybase SQL Anywhere 9

In addition, the Enterprise edition includes a full version of VCL for the Web, with unlimited access to application-mode web applications for stand-alone, ISAPI- and Apache-based applications.

The Enterprise edition adds high level modeling support through the Unified Modeling Language (UML) for both language neutral models and code-based models. Enhanced with LiveSource™ the two-way model to code / code to model feature for Delphi developers gives extended support for keeping models and code synchronized. Documentation can be automatically generated providing an easy way to explore and review projects, and supporting communication beyond the core team of developers.

DataSnap technology continues to evolve as demands for distributed computing increase. The technology behind DataSnap has moved beyond the approach of remoting data through Microsoft's COM/DCOM to a more open communications approach based on TCP/IP. This evolution has allowed the DataSnap technology to expand its capabilities to include a complete middleware technology. One of the key features behind the technology is that it is fast; fast to build, fast to deploy, and fast to execute in production.

The Enterprise edition is attractive to corporate developers who need access to corporate data, ISVs who write applications that require heterogeneous database support, as well as consultants and VAR that support Enterprise level developers.

ARCHITECT

The Architect edition of RAD Studio 2010 includes everything in the Enterprise version, as well as data modeling and design capabilities from Embarcadero ER/Studio Developer Edition. Embarcadero ER/Studio is an industry-leading data modeling tool which helps companies discover, document, and re-use data assets. With round-trip database support, data architects have the power to easily reverse-engineer, analyze, and optimize existing databases. Productivity gains and enforcement of organizational standards can be achieved with ER/Studio's strong collaboration capabilities.

The Architect edition is of interest to the same developers who are interested in the Enterprise Edition, but who want to do database design and management in a highly productive environment.

INSTALLATION

For the internet download (or Electronic Software Delivery, ESD), the customer receives a small launcher that, when run, downloads the necessary binaries from the internet and installs RAD Studio 2010. Additional items are available for download on the Registered Users web page at http://cc.embarcadero.com/reg/rad_studio.

For users who wish to install via DVD or create backup media, ISO downloads are available. A media kit DVD is also available for purchase from the Embarcadero online store at http://shop.embarcadero.com as well as from Embarcadero partners.

THE INTEGRATED DEVELOPMENT ENVIRONMENT (IDE)

When RAD Studio 2010 is run, the developer is presented with the Integrated Development Environment, or IDE. The IDE brings together in a single application all the features that a developer needs to develop applications. Encompassing an Editor, a Form Designer, a Project Manager, a Debugger, and numerous other features that enable developers to develop applications quickly and easily, the IDE allows developers to do all there work in a single environment that ties together all the functionality they need.

GENERAL NOTES

The IDE provides a user interface familiar to Windows users. Basic functionality is provided via drop-down menus and configurable toolbars holding tool buttons. Many of the various windows in the IDE are dock-able, allowing a developer to fully customize their working environment.

Desktop layouts can be saved. Desktops can be assigned for specific purposes such as debugging. Applications can be run and debugged right in the IDE. Developers can set options for almost any aspect of the IDE and their applications and projects. The entire IDE is designed to be customizable and to enable efficient and fast development.

SPEED IS THE KEY – THE NEW IDE INSIGHT

A paradigm shift is in the works with the new IDE Insight feature found in RAD Studio. This allows developers to quickly find anything in the IDE like from the projects, the components, templates, or configurations settings. By simply pressing the F6 key, the IDE insight is activated, and then the developer starts to type in the desired feature or function they want to perform. A list of possible items is displayed. The developer can then select the feature of function and it will be executed or the IDE will go to that location. IDE insight is available throughout the entire IDE making it on a key-press away at any time.

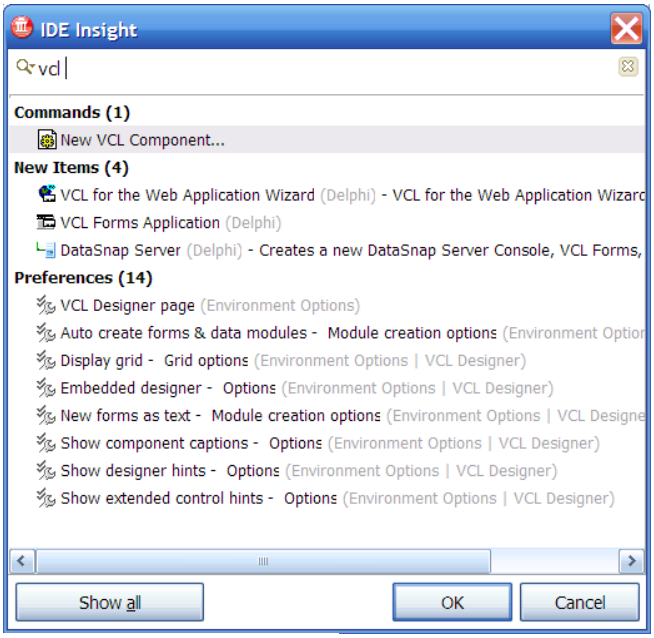


Figure 1 -- IDE Insight

For the developers who are heavy keyboard users, the IDE Insight can also be started by pressing the CTRL and the '.' Keys.

So a quick example of how you could use IDE Insight would be:

1. Start the IDE
2. F6 – IDE Insight
    a. Type 'vcl forms (pick new application)
3. F6 – type "Tedit1" and hit enter
4. F6 – type "Tlistbox1" and hit enter
5. F6 – type "Button" and hit enter
6. Now align the components on the form

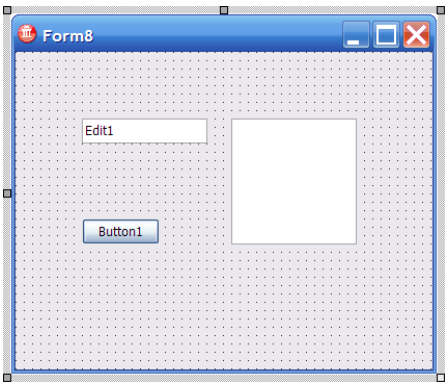


Figure 2 -- Example Application

7. Go to code – double clicking on the button
    a. Hit the F6 and show the templates
    b. Cancel – hit the ECS key
8. Put in the line of code
    a. ListBox1->Items->Add(Edit1->Text);
9. Run  on the toolbar

The above example would basically take the contents found in the Edit1 box and put them into the ListBox when the Button is pressed.

The speed in which the applications can now be created is great, and the ability to find almost anything in the IDE is an incredible help to the developer. Plus, with the added bonus of having the ability to extend the IDE Insight when the underlying IDE gets extended is an additional value.

CLASSIC APPROACH TO DEVELOPMENT

Some developers really enjoyed the old Delphi 7 and C++Builder 6 approach to IDE layout. The old products only supported a Single Document Interface (SDI) and it worked well for multiple monitors. Ever since those releases, RAD Studio has been based on Multiple Document Interface (MDI) and has supported a limited classic mode, however it never allowed for complete separation of the design surface. Now in RAD Studio 2010, complete support for the SDI has been enabled.

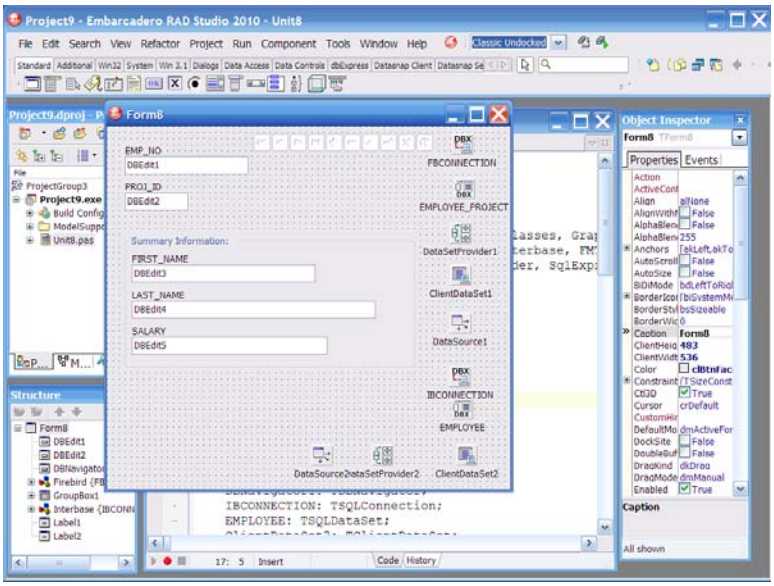


Figure 3 -- Delphi using the SDI approach

WRITING CODE -THE CODE EDITOR

No mere text editor, RAD Studio 2010's Code Editor provides extensive support for typing and numerous aids for code creation that greatly enhance a developer's productivity. The editor is hosted in a tabbed window, so it can open and edit any number of files at once. The Code Editor contains all the productivity features that developers have come to expect:

- Syntax highlighting of code makes it clear what each section of code is – comments, strings, identifiers, keywords, and reserved words are all color coded for easy identification.
- Code Completion provides hints to available identifier names as a coder types.
- With Live Templates, developers can, with a few keystrokes, invoke large chunks of code and fill in the pertinent parts of that code very quickly. Live Templates are simple XML files, so developers can quickly and easily create their own templates. Live Templates

are also completely scriptable, allowing developers to do anything they can imagine in the Code Editor.
- Class completion automatically generates implementation stubs for class declarations.
- Refactoring support helps a developer rewrite code without introducing errors to make it more readable and organized.

Figure 4 -- The Code Editor

- CodeInsight produces pop-up windows that give insight into parameters needed for a given routine.
- ErrorInsight provides immediate feedback buy underlining code syntax errors.
- HelpInsight provides pop-up windows right in the editor giving basic documentation and declaration information about any identifier within code.
- Block Completion ensures that all code is properly opened and closed. For instance, when a developer types a `begin` and then hits the `Enter` key, the corresponding `end` is automatically added, ensuring that code is properly formatted without interrupting the developers flow of concentration.
- Navigation throughout code is made easy by "context-sensitive" code. Clicking on an identifier while pressing the Control key will take the developer to the declaration or implementation of that identifier. Using a stack-based model, the developer can navigate back and forth through code. Using simple keystrokes, a developer can move between the declaration and implementation of class methods. Large files can be navigated easily with its support for IntelliMouse scrolling.
- Line numbering provides immediate location information. Bookmarks can be set, allowing the developer to quickly return to specific locations in an application's code.
- When multiple lines of code are selected the SyncEdit icon becomes available in the margin. Entering SyncEdit mode provides a quick and easy search and replace through the highlighted block of text. While SyncEdit works with highlighted blocks of code, the Rename Refactoring provides a context-sensitive search and replace across the whole project, so that only identifiers that truly refer to the same entity are renamed.
- Macros can be recorded and replayed to execute common typing tasks.

- New **Code Formatting** has been added for both Delphi and C++.

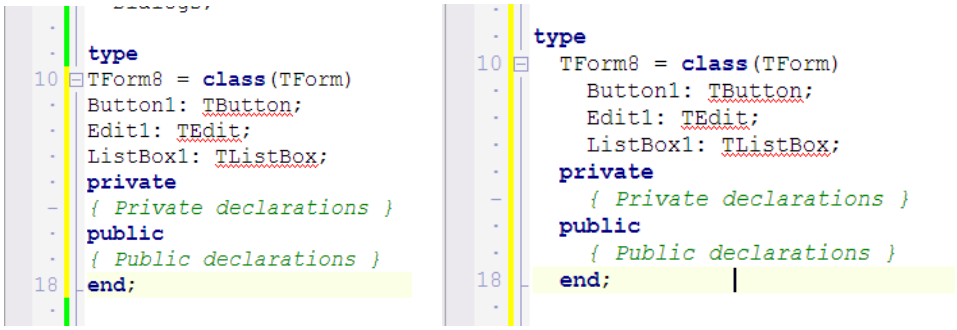


Figure 5 -- The Code Formatter

By simply hitting the Format off the Edit menu or hitting the CTRL-D the code will automatically be formatted to the format specification found in the options pane.

Overall, the Code Editor is designed to make typing code easy, efficient, and effective.

VISUALLY DESIGNING A USER INTERFACE - THE FORM DESIGNER

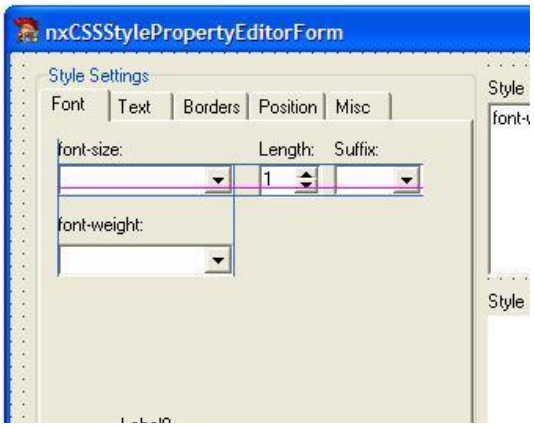
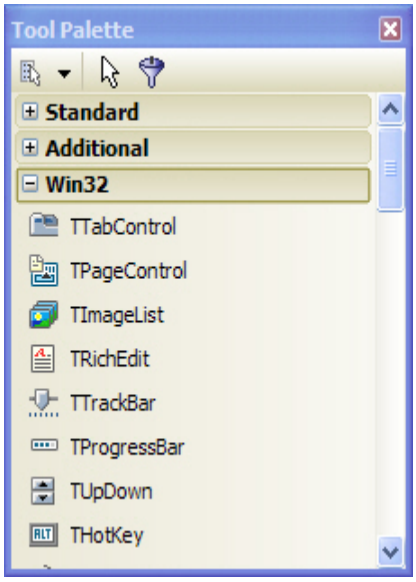


Figure 6 -- The Form Designer using Visual Guidelines to easily align controls on the form.

When not writing code, much of a developer's time will be spend laying out forms with components to create a user interface. The IDE provides a powerful Form Designer to do just that. RAD Studio 2010's Form Designer looks exactly like a Window, enabling What-You-See-Is-What-You-Get (WYSIWG) layout of forms at design-time. Developers can drag-n-drop components from the fully configurable Tool Palette and place them on the form as desired. Components can be spaced and aligned easily using the Visual Guidelines – colored lines that provide visual cues to spacing between components and alignment with other components as a component is dragged on a form.

Once components are place on the form, their properties can be set using the Object Inspector. Listing all a components properties and events, the Object Inspector allows a developer to quickly and easily manipulate a components appearance, and to attach code to the various events that might occur on that Component.

For instance, with the Object Inspector, a developer can control the position and size of any component. He can add buttons to toolbars, change text of edit boxes, and change the color of the background of a form. In addition, he can cause code to run when certain events – OnClick, OnMouseOver, OnKeyDown, etc. – occur to the component.

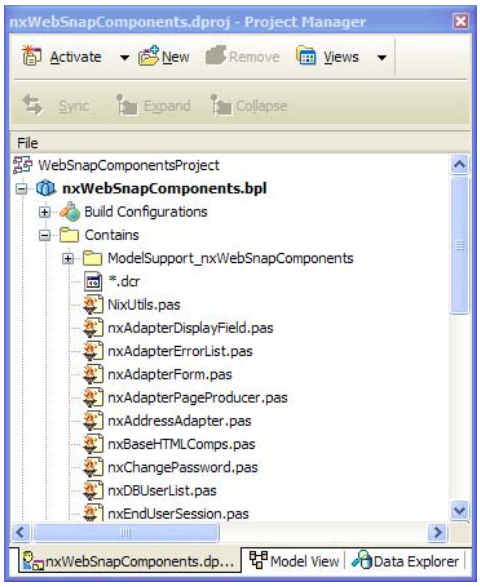


The Tool Palette houses all the VCL components installed into the IDE. It includes by default all the standard components that ship with RAD Studio 2010. In addition, developers can add in components from third-parties.

The Tool Palette is completely configurable. Components can be grouped as desired. They can be easily found using filtered searching. The layout and coloring are user-definable. Components can be selected and dropped on the form either with the mouse or keyboard.

Figure 7 -- The Tool Palette

MANAGING AN APPLICATION'S CONTENT - THE PROJECT MANAGER



Applications quickly become complicated with numerous forms and code files. Many applications consist of any number of different binaries and projects. RAD Studio 2010's Project Manager enables developers to manage their projects right in the IDE. The Project Manager organizes files and forms into projects, and projects into Project Groups. Developers can create new forms and files, add existing forms and files to a project, and add new projects to a project group. Projects can be compiled and built right in the Project Manager. Files and forms can be opened in the IDE. Projects can be rearranged so that they compile in an order dictated by the needs of the application. Everything that a developer needs to do with regard to managing the files and forms that make up a given application can be handled in the Project Manager.

Figure 8 -- The Project Manager

In addition, the Project Manager can manage multiple Build Configurations on a per project basis. It can also manage, save, and reuse specific option sets to make managing those configurations even easier.

Plus, the ability to control the Open/Reopen Project has been greatly enhanced. Now, the Tools|Options|Reopen Menu gives complete control over the number of projects and files being displayed.

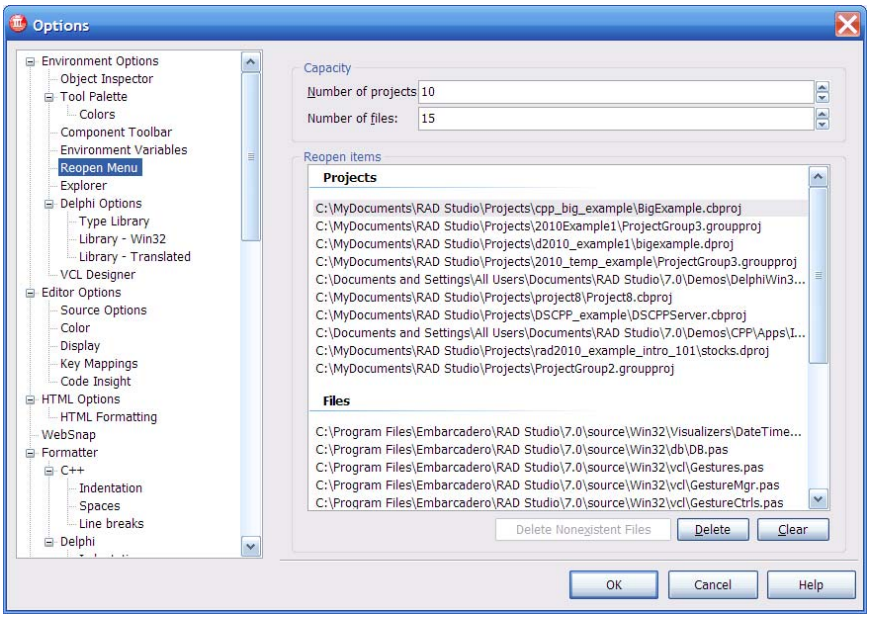


Figure 9 -- The Reopen Menu configuration page

LOOKING UNDER THE HOOD - THE DEBUGGER

All development runs into problems. When bugs or unexpected application behavior occur, a developer needs to be able to peer into the inner workings of their application, and see into what is happening as the program executes. RAD Studio 2010 provides a debugger integrated into the IDE that provides deep access and insight into an application as it runs.

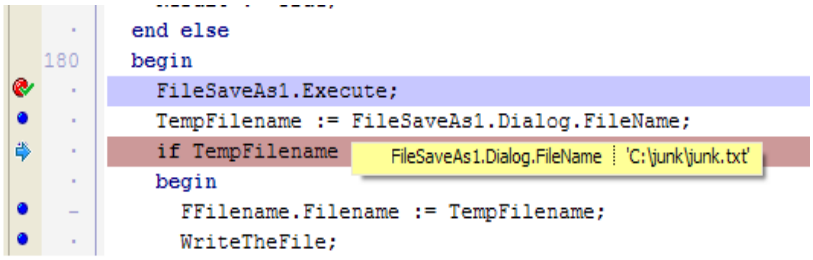


Figure 10 -- The Debugger stopped on a breakpoint

New in 2010 is the ability to have Debug Visualizers, which are special viewers of non-readable data. What is non-readable data you may ask? That may be something like TDate, TTime, or TDateTime objects, which are usually stored in a long representation. Using the new visualizer the data can be represented in human readable form.
This will help countless numbers of developers that have either custom objects or data that is not normally understandable in a binary form.

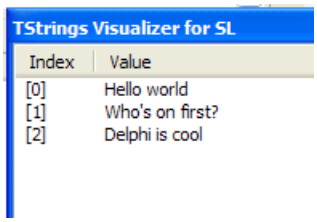


Figure 11 -- The Debug Visualizer

When an application is run within the IDE, the debugger takes over and allows the developer to control the process of execution and to gain access to all information about the entire process. Developers can set breakpoints anywhere within their code, stopping execution. Breakpoints are configurable, and can be set to trigger every time, after a certain number of times, or based upon some condition evaluating variables and functions within the users' code. An implicit breakpoint can be triggered any time an exception is thrown. Once execution is halted, the debugger will provide access to all information within scope. Developers can set watches to track the value of any variable. They can step into code, executing it line-by-line and inspecting the value of variables at anytime. The debugger displays the current call stack, all loaded modules, and the status of all threads associated with the application. If even more detail is required, the debugger can display a CPU-level view, showing the exact assembly code being executed.

Also new in 2010 is the ability to Freeze and Thaw individual threads. This is becoming more and more important as the majority of applications become multi-threaded to take advantage of the new multi-core processors.

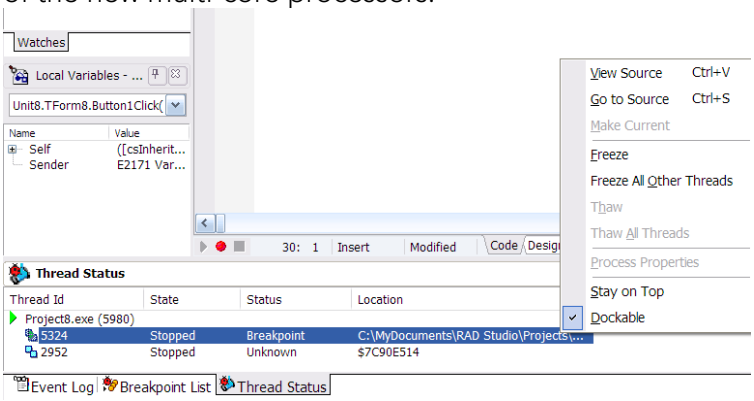


Figure 12 -- The Debug Freeze and Thaw interface

The interface also allows developer to set breakpoints on individual threads as well.

ACCESSING DATA – THE DATA EXPLORER

Many applications require access to data. The IDE includes the Data Explorer – a panel in the ID that provides quick and powerful access to database data. In the Data Explorer, a developer can create connections to the RDBMSs supported by dbExpress, RAD Studio's database access technology. Once a connection has been created, the Data Explorer can be used to browse a given database's data and metadata. Once a connection and data is available, connections and tables can be dragged from the Data Explorer to the Form Designer to create data access components on the form.

This allows new database support like Firebird to be added to the explorer, bringing all the rich tools along with it. These tables can be dragged from the Data Explorer and placed onto a Form or DataModule, which will automatically bring the connection and table component for ease of database access.

In addition, the Data Explorer can be used to examine data, and build queries using the built in Visual Query Builder.

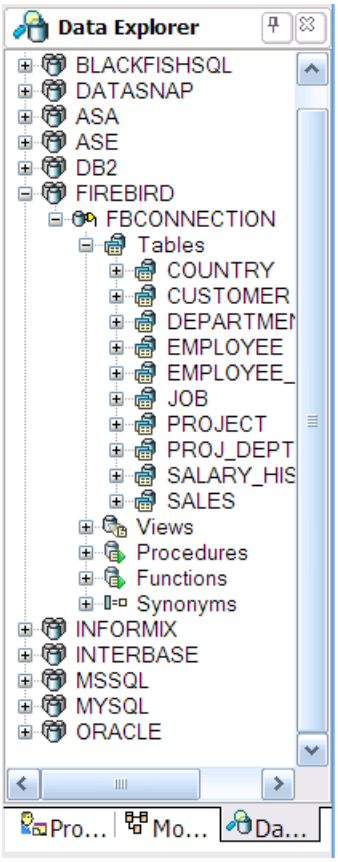


Figure 13 -- The Data Explorer

THE VISUAL COMPONENT LIBRARY

While RAD Studio 2010 can be used to build almost any kind of application, the heart of its strength lies in building windowed client or stand-alone desktop applications. For this, RAD Studio 2010 provides an application development framework called the Visual Component Library (VCL). The VCL is a class library that encompasses the wide-range of Win32 API's to encapsulate the building of Windows applications. It is component-based, meaning that it is designed to allow developers to build components that can be manipulated on the Form Designer at design-time.

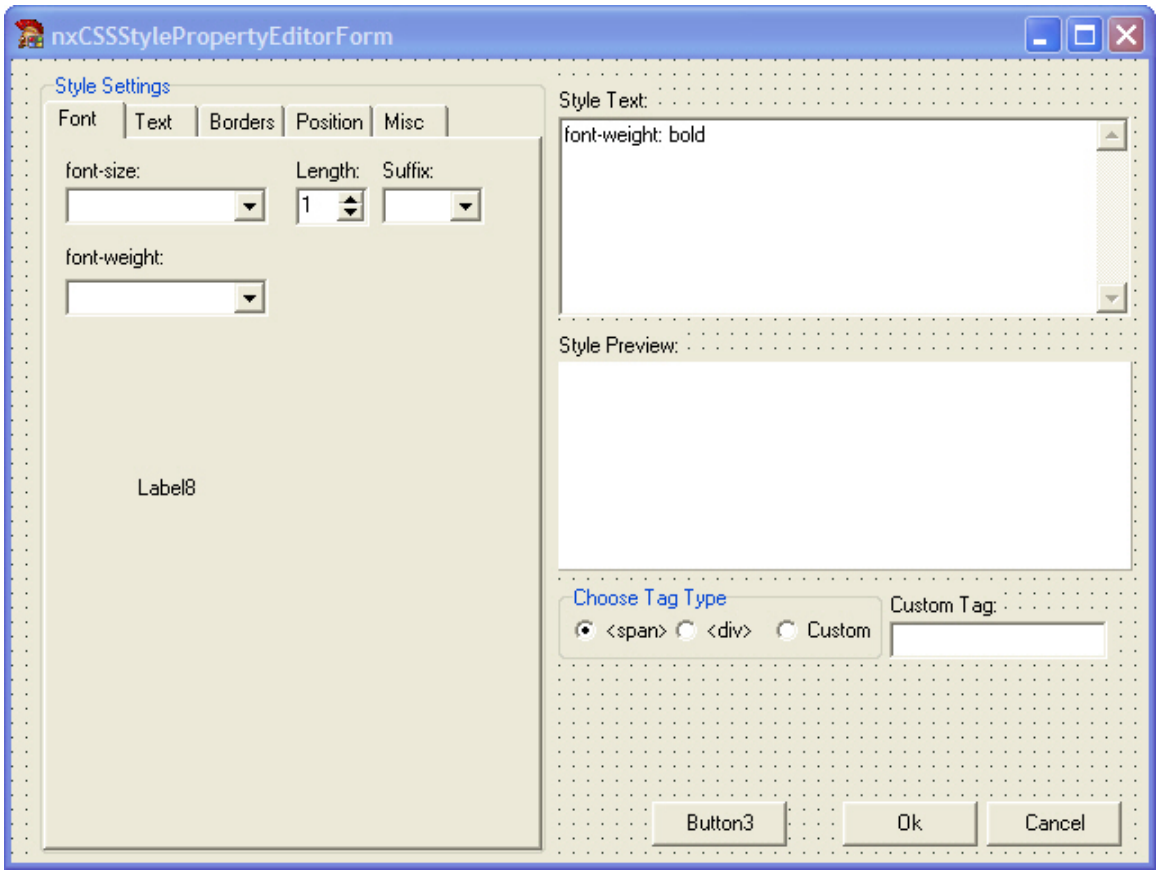


Figure 14 - A Form with a number of VCL controls on it.

The VCL is the foundation for all Windows application development in RAD Studio 2010. It provides an encapsulation of a Windows-based window in the `TForm` class. (By longstanding convention, classes in Object Pascal are pre-pended with a 'T'). The Form Designer in the IDE provides developers with a "canvas" on which to design a user interface. The VCL wraps up all the standard Windows UI controls in components such as `TButton`, `TEdit`, `TLabel`, `TCheckbox`, etc. The framework is extended by simple inheritance, allowing the users to easily enhance the IDE with their own components. (As a result, there is a rich, broad community of third-party developers – commercial, freeware, and open source – that provide a wide-range of feature-rich components for VCL developers to use.)

The VCL has proven to be remarkably robust over the years, having been adapted to numerous different platforms. The VCL started out on 16-bit Windows, but was soon moved to the 32-bit world to support Windows 95. It has even been, in the past, adapted to support Linux and .NET.

TOUCH THE FUTURE

Rapidly build touch based GUI, tablet, touchpad, and kiosk applications or easily upgrade existing applications UIs with little or no additional coding.

- Pluggable gesture engine architecture
- Works on all supported versions of Windows (2000, XP, Vista and Windows 7)
- Use touch-enabled hardware or work with what you have (e.g. mouse)
- Integrated support for touch and multi-touch interfaces in the base VCL
- 30+ standard gestures for panning, zooming, rotating and more
- Create your own with the Custom Gesture Editor
- Touch Keyboard - a complete virtual keyboard for enhanced non-keyboard interface interactions that supports multiple locales and languages

NEW DATABASE AND TOUCH FEATURES FOUND IN DELPHI AND C++BUILDER 2010 (EXAMPLE)

The following is a step-by-step project for creating a multiple database application with gesture support. For this example you need to have both Firebird and InterBase loaded onto the machine. InterBase 2009 Developer is included in the package and Firebird can be downloaded from the Firebird website: http://www.firebirdsql.org/

Note: *Make sure to have the Firebird client library (fbclient.dll) either in the path or in the bin directory of RAD Studio.*

1. Create a new VCL Forms application – F6 (follow steps above)
2. Go to the Data Explorer tab
    a. Open up Firebird connection show the tables and things
    b. Open up the InterBase connection show the tables and things
    c. Drag Employee_Project from the Firebird connection to the form and drop
        i. This will place two components on the form
            1. FBConnnection
            2. EMPLOYEE_PROJECT
    a. Drag a DataSetProvider (Data Access on the Tool Palette)
        i. Set DataSet to Employee_Project in the property editor
    b. Drag a ClientDataSet (Data Access on the Tool Palette)
        i. Set Provider to DatSetProvider1
        ii. Right-mouse click on the ClientDataSet1 component and select Field Editor
        iii. Right-mouse click and add all fields
        iv. Drag Emp_NO and Proj_ID to the form, notice that a DataSource1 is added
    c. Drop a Navigator from (Data Controls of the Tool Palette)
        i. Connect to DataSource1 in the property editor

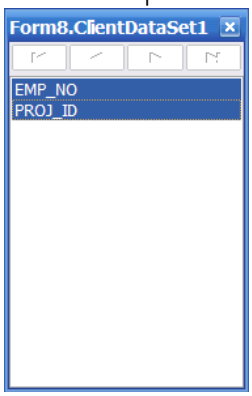


Figure 15 -- Fields Editor

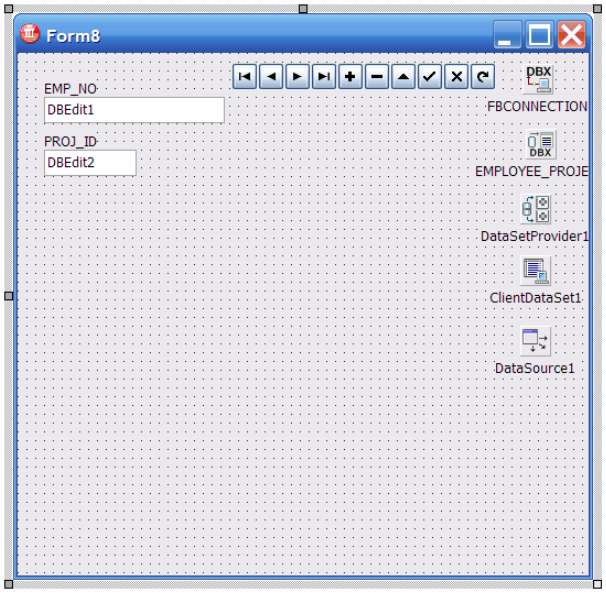


Figure 16 -- Screen layout so far

10. Open InterBase in the Data Explorer and drag Employee table onto the form
    a. This will drop two components
        i. IBConnection
        ii. EMPLOYEE
    b. Add DataSetProvider2 (Data Access on the Tool Palette)
        i. DataSet = Employee
    c. Drop a TClientDataSet2 (from Data Access on the Tool Palette)
        i. ProviderName = DataSetProvider2
        ii. MasterSource = DataSource1 (Firebird database)
        iii. MasterField = Emp_No = Emp_No

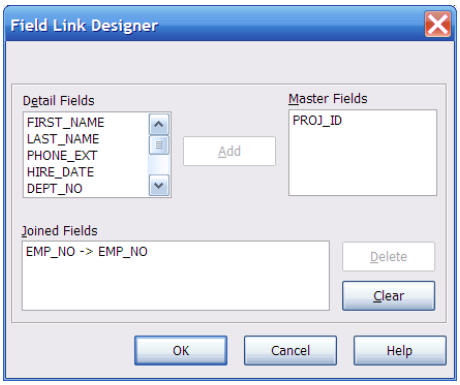


Figure 17 -- Setting a mater-detail structure

    d. Drop a GroupBox from the (Standard Tool Palette)
    e. Right-mouse click on the ClientDataSet2 select Field Editor
        i. Add all Fields
        ii. Drag only
            1. First_Name
            2. Last_Name

3. Salary to the form

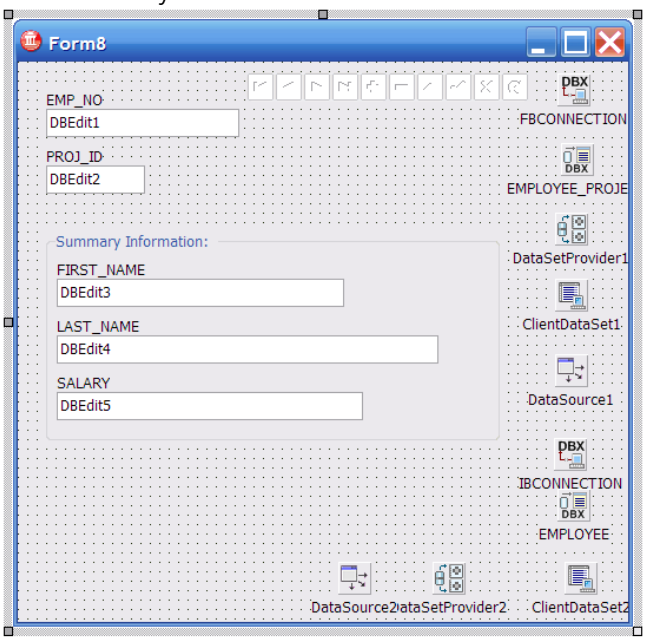


Figure 18 -- Setting a mater-detail structure

iii. Set EMPLOYEE_PROJECT, EMPLOYEE active setting to True
iv. Set ClientDataSet1, ClientDataSet2 active setting to True
    1. You should see live data in design
v. Save All
vi. Run 

GESTURES / TOUCH

Now we want to add gesture support to the above application. Since both touch and gesturing are now built into the VCL, all VCL applications can take advantage of this excellent feature. Even users without touch screen computers can use the gesture support via a mouse.

11. Adding Gestures
    a. Add ActionManager from (Additional from Tool Palette)
        i. Right-mouse click on ActionManager and Customize

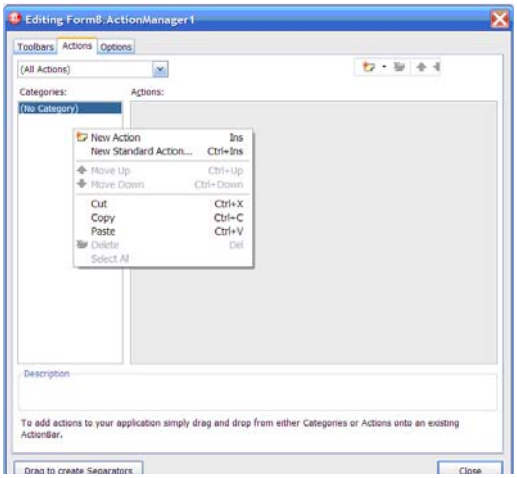


Figure 19 -- Customizer for ActionManager

ii. Select New Standard Action…
iii. Go to DataSet actions

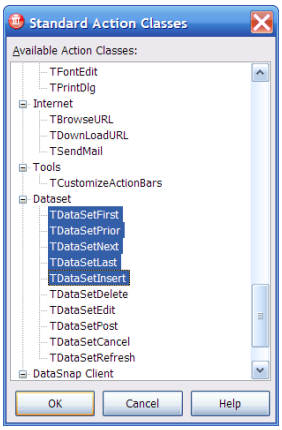


Figure 20 -- The Actions we want to support with gesturing

iv. Select First, Prior, Next, Last, and Insert
v. Click the OK
vi. Show the DataSet in the ActionManager
vii. Click the close button

b. Quick discussion on where is the Gesture allowed (individual controls or form)
   i. Highlight Touch on form
   ii. Highlight Touch on Edit box
c. Drop Gesture manager on the form
d. On the Form set the Touch to Gesture1
   i. Right-mouse click on the Gesture1 component and show Customization
e. Back to the form and touch
   i. Show the default gestures under the Gesture (under standard)
   ii. We only want to set Next and Prior
      1. LefttoRight = Action Next
      2. RighttoLeft = Action Prior
f. Save all
g. Run

When the application starts you should be able to hold down the left-mouse button and drag it across the screen from left to right to go to the next record, which is the same as pressing the > button on the DBNavigator component. Likewise you should also be able to hold down the left-mouse button and drag it across the screen from right to left to move to the prior record.

NEW CLASS EXPLORER FEATURE FOR C++BUILDER

C++Builder 2010 also features a new Class Explorer. The Class Explorer is built on class modeling functionality and provides the ability see the class structure within an entire project. Developers can choose to see classes in a top-down or bottom-up view. In addition, the Class Explorer can be used to add classes, methods, and procedures to an existing class library. Plus, it is incredibly fast.

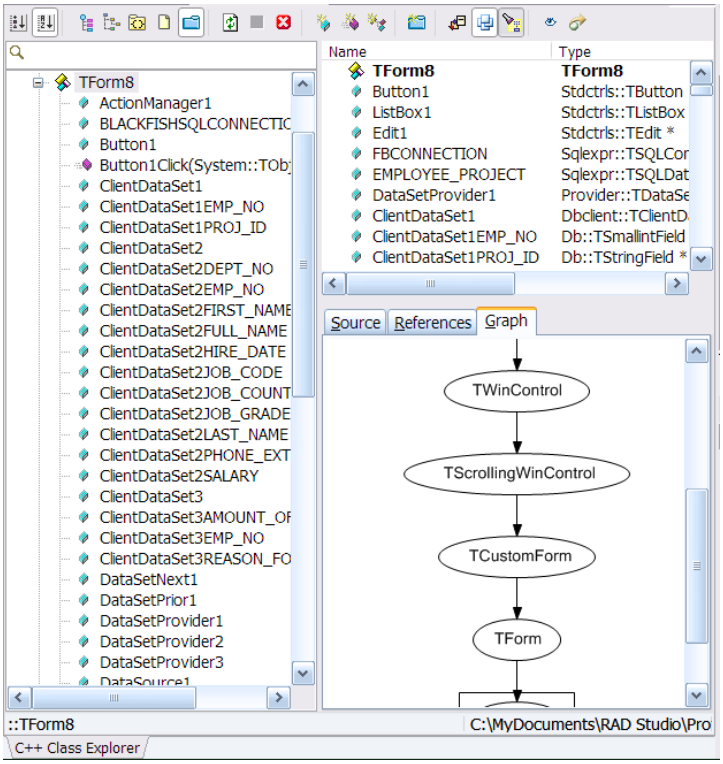


Figure 21 -- New C++ Class Explorer interface

NEW MULTI-TIER DATABASE ARCHITECTURE – DATASNAP 2010

Delphi was one of the first development tools to include a multi-tier database development framework called DataSnap. It allows developers to build middle-tier application servers that serve data and manage business rules in a single application. The middle-tier is the "sentry" to the database, providing access to the data and enforcing business rules on the processing and updating of that data. DataSnap also provides a powerful client solution which provides access to the middle tier, as well as a powerful in-memory dataset for managing and manipulating data on the client.

RAD Studio 2010 includes a major update to the DataSnap architecture. In previous versions, DataSnap made use of COM technology. In RAD Studio 2010, those dependencies are removed and replaced with a powerful yet lightweight implementation called Server Methods. Server Methods allow the developer to write methods that are part of the middle-tier. Those methods are then made available to the client in a seamless way. Developers can call server methods exactly as if the code were executing within the client binary. Server methods can pass, as parameters, any type from the dbExpress type system, including strings, integers, datasets, datareaders, connections, and OLEVariants. This makes for a very powerful means of passing data between client and middle-tier.

The new DataSnap also provides support for existing DataSnap servers.

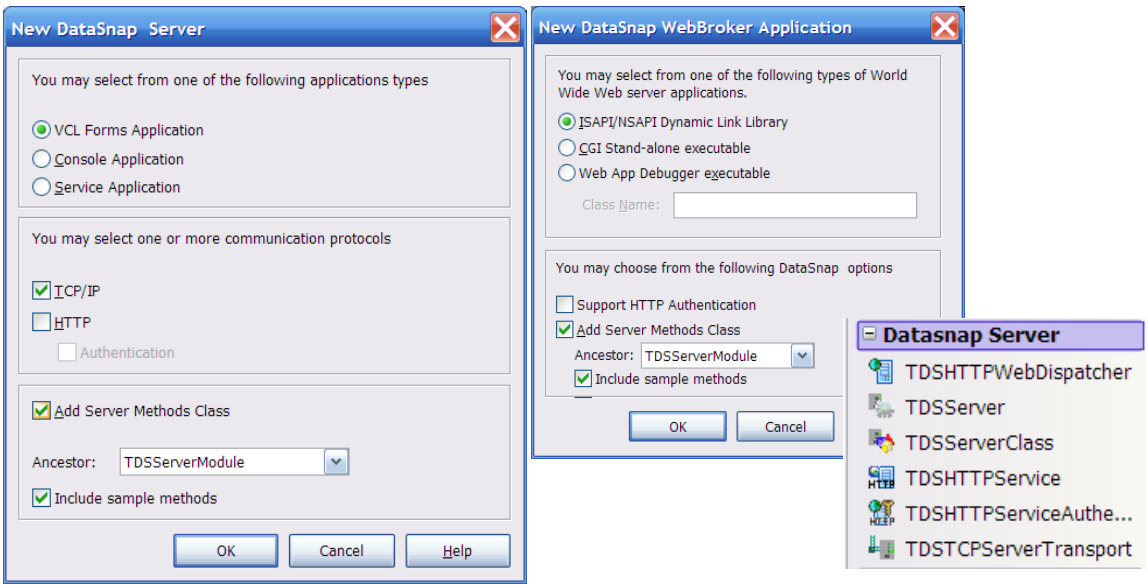


Figure 22 -- New tooling for DataSnap

The new DataSnap supports HTTP/HTTPS access, Restful Web Services access and the ability to add Authentication and Encryption. Plus, it allows the in-process deploy to an IIS server.

FULL UML INTEGRATION WITH AUDITS AND METRICS

A model view can be reverse engineered directly from the projects. This creates a hierarchical Model View where the project classes are presented in a tree, and also can be represented as UML class diagrams. This model and the diagrams can then be used to automatically generate documentation for the project.

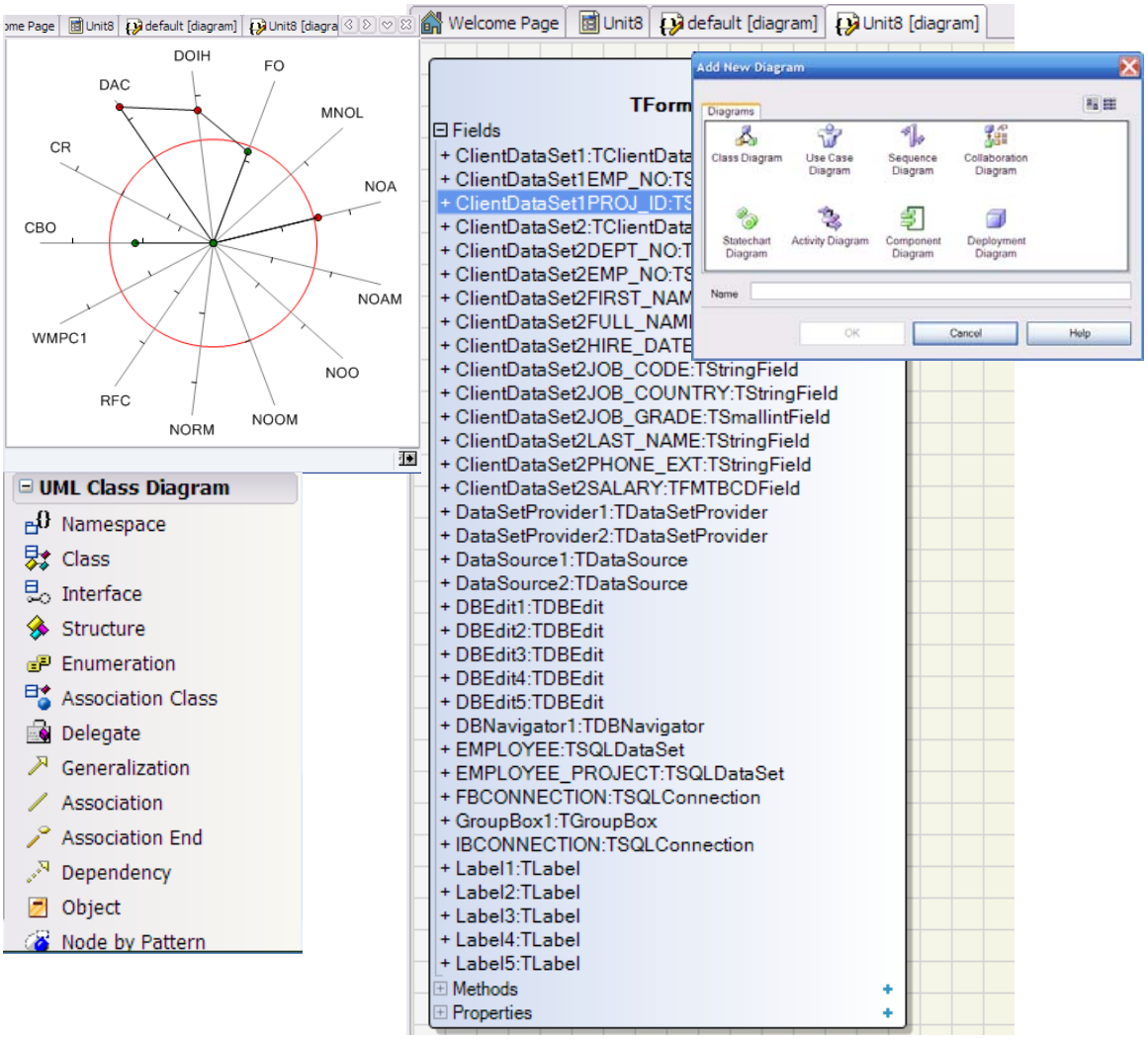


Figure 23 -- UML class diagram generated from a Model View

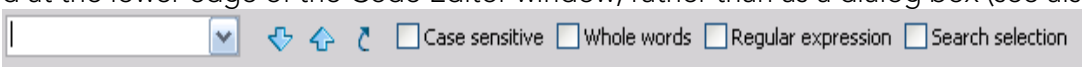
The UML integration supports both code-based models and non-code-based models. That means a developer can create models with no particular language in mind, and then RAD Studio can take those non-language models and generate source code from them. The package also includes full source code Metrics and Audits that allow you to check the health of the object code and find poor coding practices.

NEW RTTI AND RTL SUPPORT

Runtime Type Information (RTTI) is a programming paradigm in which information about a type can be obtained at run time. If RTTI generation is enabled, the resulting binary includes special metadata that contains information about types (for example, class ancestry, declared fields, annotated attributes). Using the functionality provided in the RTTI unit, you can obtain this information at run time. The net result is the ability to create more abstract and generalized frameworks that can operate on any type that exposes RTTI.

SEARCH COMMAND CHANGES

Various Search commands have been enhanced and extended as follows:

- The **Search > Find** command (^F) has been redesigned and now appears as a **task bar** located at the lower edge of the Code Editor window, rather than as a dialog box (see also Find): 

- The IDE now highlights all Search matches. The first match location is highlighted in one color, and all the other onscreen locations of the search item are highlighted in a second color.
    - The two colors that are used are predefined for the IDE's color schemes.
    - You can customize the colors by selecting background and foreground colors for the **Additional search match highlight** element on **Tools > Options > Editor Options**.
    - To disable the highlighting of all search matches, uncheck **Show all search matches** on **Tools > Options > Editor Options**.
- **Incremental Search** also has a new search bar:  You can just start typing, or select from previous search strings that match what you type.
- The **Search > Find in Files** dialog box has a new field (**Directories**) that accepts wildcard specifications such as *.pas or *.cpp. You can also specify multiple directories -- either by separating directory names with semicolons in the **Directories** field or by clicking the new **Folders and Groups** button to open the Select Directories dialog box. On **Select Directories**, you can construct directory lists and directory groups. For more information, see:
    - Find in Files
    - Searching in Directory Groups
    - Searching in a List of Directories
    - Select Directories

NEW FEATURES INTRODUCED IN RAD STUDIO 2009

To a reviewer, the most interesting features in an upgraded product are those that are new in the latest version. This section will give a look at the features that were introduced in RAD Studio 2009, many of which have been expanded or extended in RAD Studio 2010.

DATABASE DESIGN AND MODELING

The Architect edition of RAD Studio includes a complete solution for designing and modeling databases. Developers can use the included version of ER/Studio Developer Edition to create either entity-relationship or physical models of a database. They can then export that model to any of the multiple databases supported by ER/Studio Developer Edition.

An online demonstration of the features of ER/Studio Developer Edition can be found on the Embarcadero Developer Network:

http://windemo1.Embarcadero.com/Tiburon/LaunchReplays/ERStudio/ERStudio.html

NEW DELPHI LANGUAGE FEATURES

GENERICS

Generics allow the developer to write code that refers to a type without having to specify the specific type of that type. Often called Parameterized Types, generics provide developers with the ability to write general, or "generic' classes that operate on a non-specific type. The class use case for generics is a list, where the type of the items contained within the list need not be specified when the list is written.

```
10
   type

      TControlReporter<T: TControl, constructor> = class
      private
        FInternalType: T;
      public
        constructor Create;
        procedure ProcessComponent;
        procedure SetType(aT: T);
20    end;
```

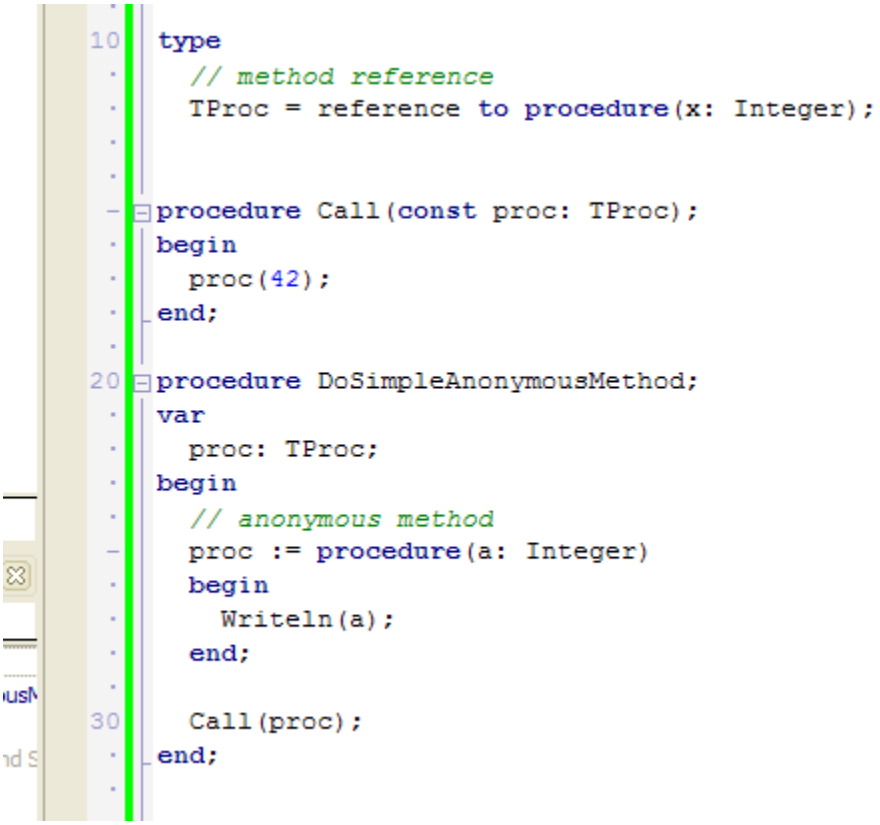
Figure 24 - A Generic class with constraints declared in Delphi

RAD Studio provides complete support for generics, including providing generic methods and constraints on generics. Constraints are the ability to limit a generic type to a specific set of functionality. For instance, a constraint might limit the generic type to only those with constructors or to those that implement a specific interface.

RAD Studio also provides new runtime library support for generic lists, collections, stacks, queues, etc.

ANONYMOUS METHODS

Anonymous Methods are code constructs that allow developers to pass code blocks as parameters. They are a procedure or function that does not have a name associated with it. An anonymous method treats a block of code as an entity that can be assigned to a variable or used as a parameter to a method. In addition, an anonymous method can refer to variables and bind values to the variables in the context in which the method is defined. Thus, Delphi's Anonymous Methods are full closure types, as they capture state when code blocks are passed.



```
type
  // method reference
  TProc = reference to procedure(x: Integer);


procedure Call(const proc: TProc);
begin
  proc(42);
end;

procedure DoSimpleAnonymousMethod;
var
  proc: TProc;
begin
  // anonymous method
  proc := procedure(a: Integer)
  begin
    Writeln(a);
  end;

  Call(proc);
end;
```

Figure 25 -- A code snippet that demonstrates anonymous methods

NEW VCL FEATURES

RIBBON CONTROLS

RAD Studio contains a full implementation of the Office 2007 User Interface controls, or "Ribbon Controls". Built purely in Object Pascal using existing VCL architecture, developers can use ribbon controls to build modern, powerful, easy-to-use GUI applications with no coding. Because they are built upon Delphi's powerful TActionManager technology, existing applications can fairly easily be migrated to use this new interface design paradigm.

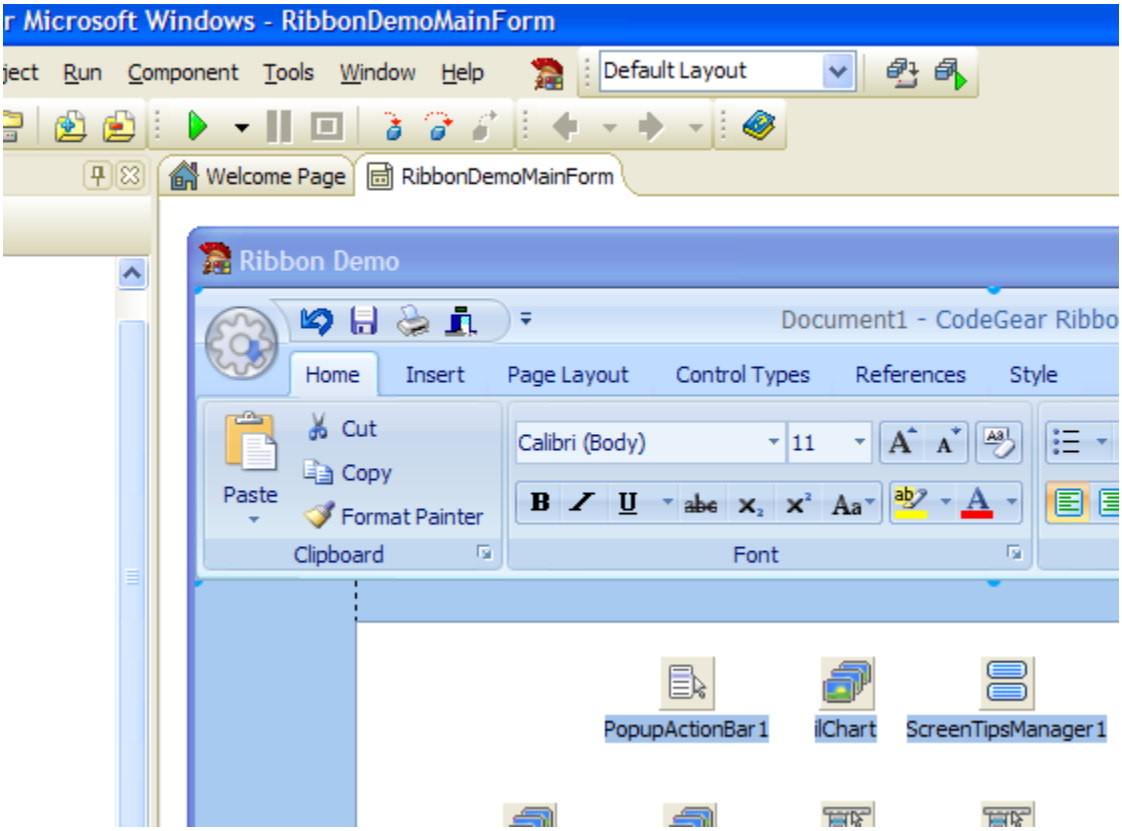


Figure 26 -- A Delphi form using the Ribbon Controls in the Forms Designer

ADDITIONAL NEW COMPONENTS

TPNGImage

RAD Studio provides full support for displaying PNG (Portable Network Graphics) images. PNG images are now supported in the TImage and TImageList components. Developers can display PNG images on their forms, or use them on toolbars, menus, and buttons.

TCategoryPanelGroup

The TCategoryPanelGroup component is a collection of collapsible panels in a single component. Similar to the notion of the "Outlook Toolbar", a TCategoryPanelGroup can contain any number of collapsible panels, which in turn can hold any number or type of other components.

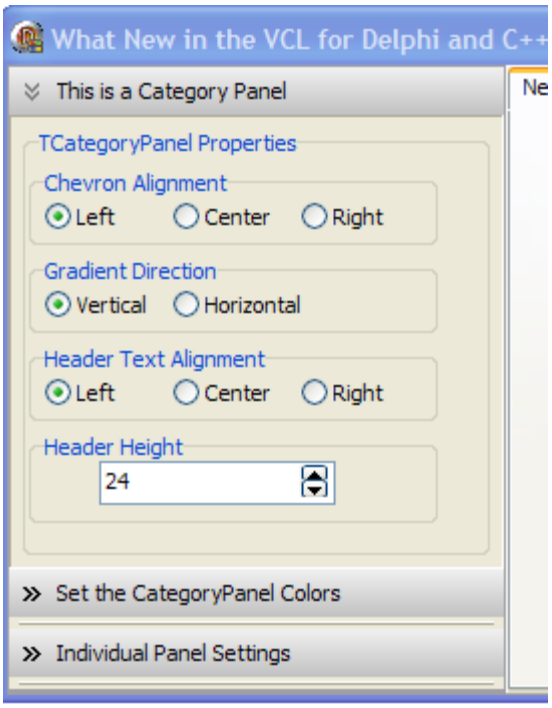


Figure 27 -- A TCategoryPanelGroup showing three CategoryPanels, two of which are collapsed.

TBallonHint

RAD Studio supports a flexible, configurable control hint system. Custom Hints can now be more easily created by descending from the TCustomHint class. RAD Studio

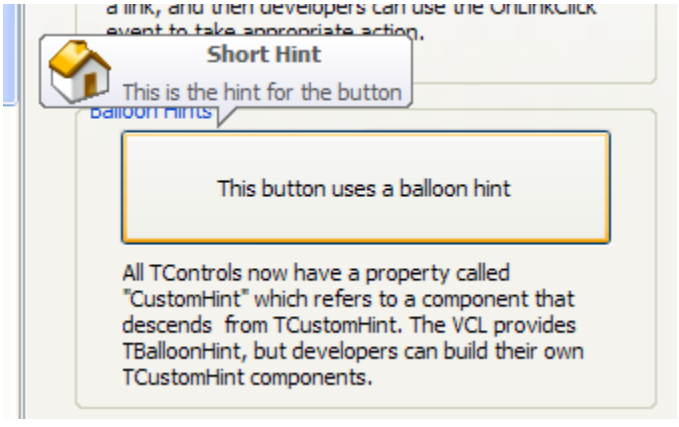


Figure 28 -- a VCL control displaying a balloon hint.

provides TBalloonHint as a default implementation. All VCL components now have a property called CustomHint which refers to a component that descends from TCustomHint.

Balloon hints can have a title and main text. They also can display images within the hint.

TButtonedEdit

TButtonedEdit is an extended edit control that allows the developer to put glyphs inside itself on both sides of the control. The controls provide events that allow code to be executed when either glyph is clicked.

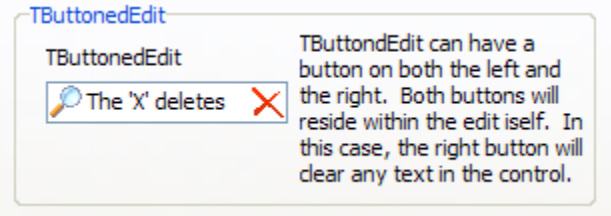


Figure 29 -- A TButtonedEdit used for searching. The Red X can be used to clear the text box.

UPDATES TO EXISTING COMPONENTS

The VCL also contains a number of updates to existing components:

- The TImageList component can now contain any image type supported by the TImage component framework
- TButton has been enhanced to support the placing of images on buttons, including having those images match the state of the button. On Vista, TButton now supports the CommandLink and SplitButton styles.
- TTreeView now supports expanded images – that is, a node can display a different image when expanded or collapsed.
- TListView now supports groups for Vista
- TRichEdit now supports the Windows RichEdit 2.0 specification
- TProgressBar now supports a themed look, the Marquee and Smooth mode styles, and on Vista, the Pause and Stopped states are supported.
- TEdit now supports customizing the Password character, and provide a TextTip property. TextTip displays "hint" text when the TEdit is empty and doesn't have the focus.

A complete demonstration of the new features in the VCL can be found on the Embarcadero Developer Network:

http://windemo1.Embarcadero.com/Tiburon/LaunchReplays/DelphiVCL/DelphiVCL.html

NEW IDE FEATURES

RESOURCE MANAGER

Most Windows applications contain Windows resources: Bitmaps, cursors, fonts, and other data that can be contained inside of a compiled binary. RAD Studio provides a Resource Manager that enables developers to easily add and manage the resources in their projects. Developers can add all of the standard Windows Resource types, given them names, and then extract them in code using the TResourceStream class.

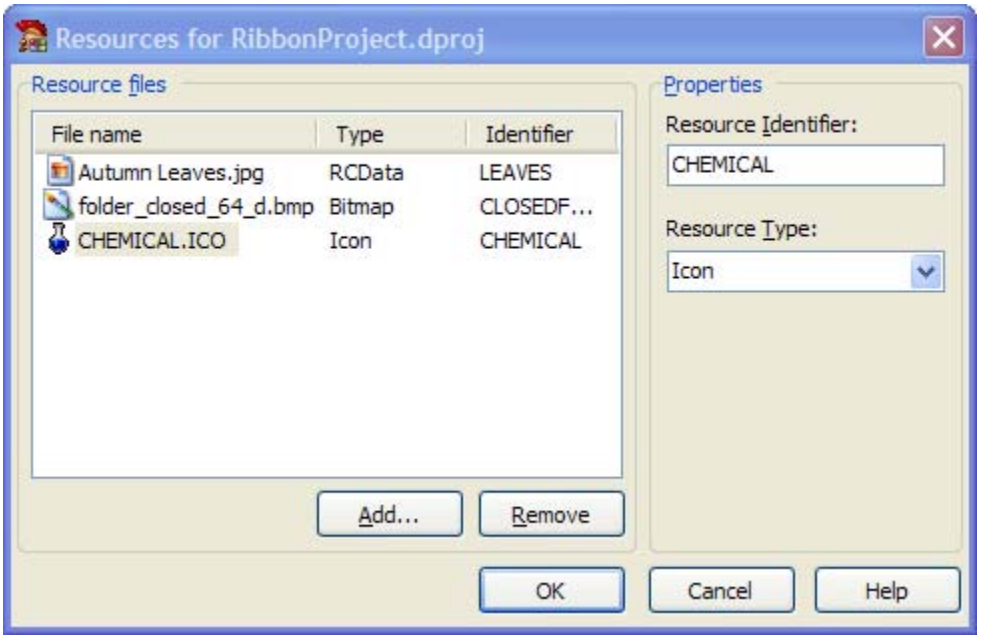


Figure 30 - The Resource Manager

BUILD CONFIGURATIONS

Many Delphi projects include numerous individual projects combined to create a single application. Developers can put together DLL's, EXE's, and packages to create a unified solution. Managing all the different projects and code files can be difficult. It can be especially difficult at Build-time.

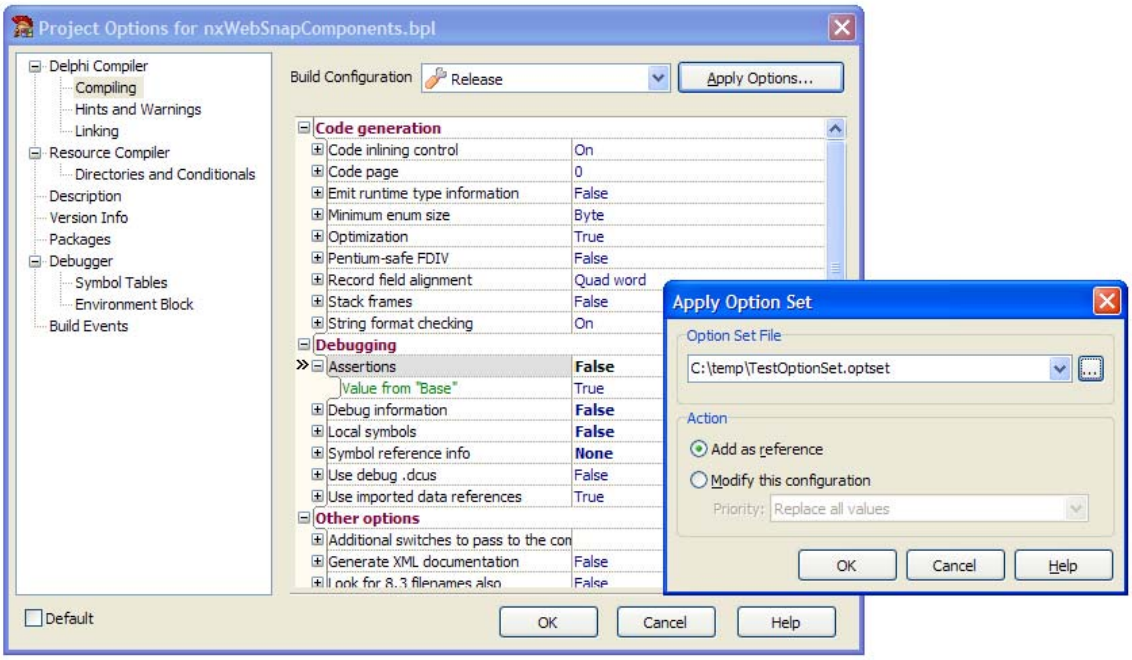


Figure 31 -- The Project Options dialog with inherited settings and applying an option set

Projects often need to be able to be built in different ways for different purposes. Builds for testing, debugging, field testing, and releasing applications might be necessary. Each different build requires a different configuration and a different set of compiler options and settings. RAD Studio provides a flexible build configuration management system which works the same in both the IDE and at the command line.

RAD Studio allows developers to create project option sets for organizing and managing build options. Option sets can "descend" from others, inheriting and overriding settings as desired. Option sets can be saved in files and applied to specific projects or other configurations.

CLASS EXPLORER

The Class Explorer for Delphi was introduced in RAD Studio 2009. The Class Explorer is built on Delphi's class modeling functionality and provides the ability see the class structure within an entire project. Developers can choose to see classes in a top-down or bottom-up view. In addition, the Class Explorer can be used to add classes, methods, and procedures to an existing class library.

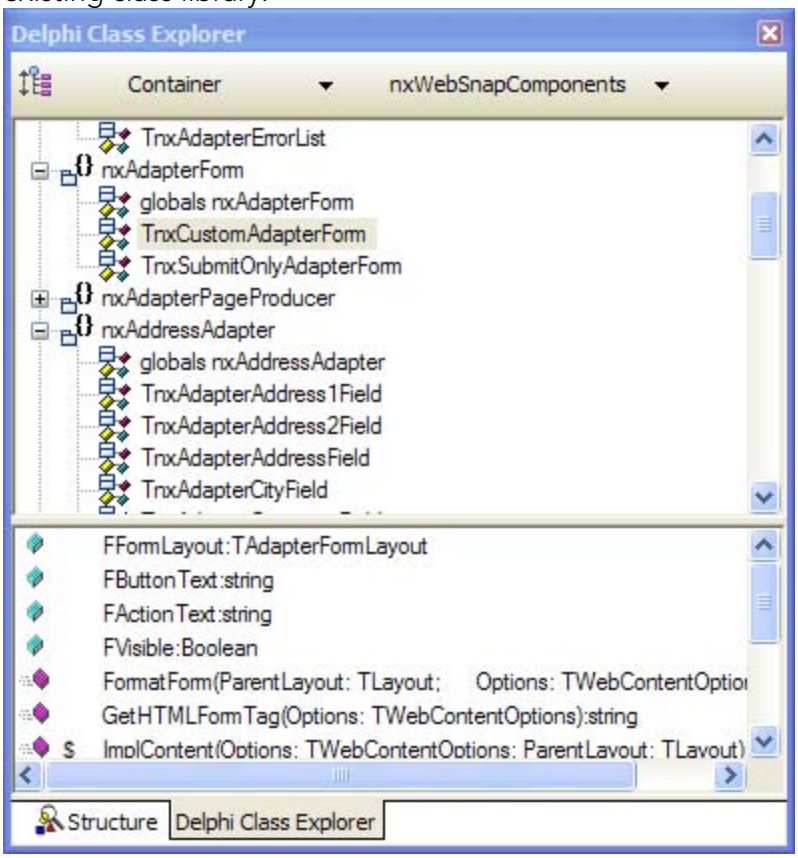


Figure 32 - The Class Explorer

TRANSLATION TOOLS

Globalization is an important theme in RAD Studio, and to take advantage of the new globalization features, developers are going to want to translate their applications into other languages. Therefore, RAD Studio 2010 includes updated and improved versions of the Integrated Translation Environment (ITE) and the External Translation Manager (ETM).

The ITE is a tool built into the IDE that enables the developer to easily product translated versions of an application. The ITE allows for the selection of any number of language projects. It then scans the main project and extracts all strings, captions, and text that can be translated. It then provides a tool built into the IDE in which the developer or translator can translate those strings. The translations are then placed into a resource DLL in the proper project. Once the resource DLL is present, it will display the properly translated strings when that project is run.

The ETM is a redistributable tool that can be sent to translators or translation services, along with the output of the ITE. This enables developers to manage easily the outsourcing of the translation of applications.

A demonstration of the new IDE features including Build Configurations and the Resource Manager can be found on the Embarcadero Developer Network:

http://windemo1.Embarcadero.com/Tiburon/LaunchReplays/Delphi2009IDE/Delphi2009IDE.html

A demonstration of the Integrated Translation Environment and the External Translation Manager can be found on the Embarcadero Developer Network at:

http://windemo1.Embarcadero.com/Tiburon/LaunchReplays/ASCIInew/ASCIInew.html

UPDATED AND IMPROVED COM/ACTIVEX SUPPORT

RAD Studio includes a COM and ActiveX development framework that was completely re-architected in the 2009 version. This new framework is based on a new "Reduced IDL" language which is a subset of the Microsoft IDL specification. It is designed specifically to define COM objects in Delphi. The combination of a RIDL file and an associated PAS file means that COM and ActiveX objects can now be defined purely in text. The two files can then be compiled into a type library file (*.TLB) which is external to the actual project itself. This means that the TLB file is a result of the compilation of the project and not part of the project itself. In this way, Delphi COM and ActiveX objects can be properly managed, merged, and stored in a source code management tool. No longer is the TLB file needed to store information about the COM/ActiveX project.

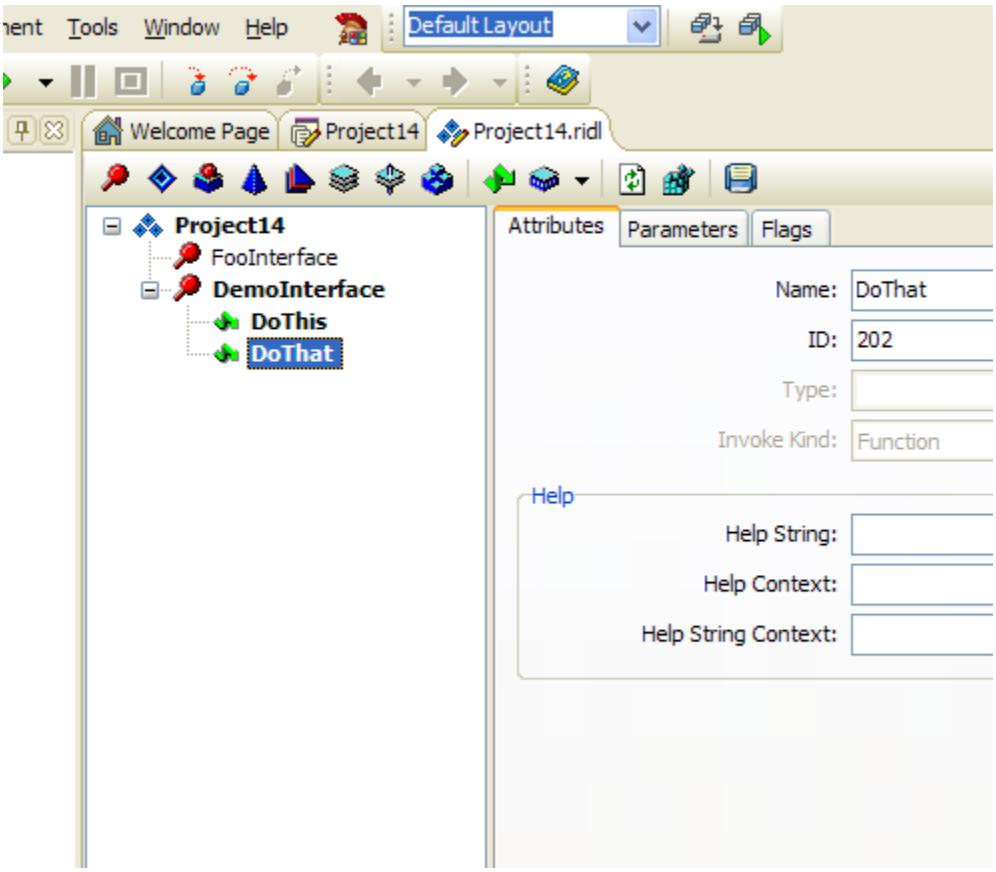


Figure 33 -- The Type Library Editor editing a *.RIDL file

A demonstration of the new COM and ActiveX features of RAD Studio can be found on the Embarcadero Developer Network at:

http://windemo1.Embarcadero.com/Tiburon/LaunchReplays/COM/COM.html

Features That Were New in Delphi 2007

The following features were new in the Delphi 2007 release. All of them, of course, are still present in RAD Studio 2010.

Blackfish™ SQL

Delphi 2007 introduced Blackfish SQL Delphi Edition. Blackfish SQL is a managed code, SQL-92 compliant relational database management system. Blackfish is a very flexible RDBMS. It can be used as an embedded database in stand-alone applications, or it can easily scale up to a full-blown enterprise database solution. It can run in-process with an application or web solution, or as a server or Windows Service. Deployment is dead simple, requiring nothing more than an XCOPY deployment including the binaries, database files, and license file.

By leveraging the .NET Framework, Blackfish SQL provides all the capabilities of a full-fledged RDBMS, yet offers the flexibility of scaling from an embedded system all the way up to an enterprise system.

Blackfish can actually be run three different ways:

1. As a Windows Service
2. As a standalone executable
3. As an in-process assembly

An application can connect to Blackfish either remotely or locally. When connecting remotely, the connecting application will use the TCP/IP stack to pass information. When connecting locally, the connecting application will simply bind to the Blackfish assembly like any other.

Upon installation, Blackfish will be installed as a Windows Service, set to automatically run on startup. This will make Blackfish universally available on a developer's machine. Blackfish SQL uses port 2508 by default.

Additional Blackfish SQL Resources

- More information about building and deploying applications can be found in this article: Developing and Deploying with Blackfish SQL and Delphi
- An online demonstration of Blackfish SQL's capabilities, done by one of the developers of Blackfish, can be found here.
- Developing and Deploying with Blackfish SQL and Delphi
- The Blackfish SQL Developers Guide can be downloaded here.
- Blackfish's Lead Developer, Steve Shaughnessy, wrote a blog post about Blackfish here.
- Leonel Togniolli, R&D Developer for Blackfish SQL, gave a presentation called Introduction to Blackfish SQL

Vista Support

The release of the Vista operating system brought a slew of new APIs that provide support for the new features in Vista. Delphi 2007 was the first tool to take direct advantage of these new Vista APIs. By wrapping the new features up as VCL components and properties, Delphi 2007 gave developers immediate and easy access to much of Vistas new functionality. RAD Studio

2010 takes this a step further by adding new Vista API support into many of the components in the VCL.

Of course, only applications built with the Vista support will only function completely as expected on Vista. When run on an XP or Windows 2000 machine, those Vista-specific features will "degrade gracefully" and appear as the basic functionality of XP.

GLASSING EFFECTS

Delphi 2007 introduced support for the Windows Aero interface. The VCL adds properties to `TForm` that create "glass" frames around the edges of a `TForm`. Native Windows developers can thus take advantage of this feature in their native applications. Implementing it is as simple as setting the `TForm.GlassFrame.Enabled` property to True. Developers can set the width of the glass frame around the outside of the form, or even set the `SheetOfGlass` property to `True` and have the whole form be "made of glass".

If an application uses the glassing effects and is run on non-Vista systems, the glassing effect does nothing.

VISTA DIALOGS

The new Vista operating system also enhances the standard dialogs that ship with the control set of the operating system. The VCL encapsulates these new dialogs in the `TFileOpenDialog`, `TFileSaveDialog`, and `TTaskDialog` components. These components encapsulate the functionality provided by the new Vista dialog controls, making it very easy for developers to incorporate them into their applications.

If an application uses these new dialogs is run on a non-Vista system and invokes the new dialogs, a special exception will be raised, as this functionality is only supported on Vista-based systems. Developers can check for this and invoke the correct supported behavior for the running operating system.

AJAX AND VCL FOR THE WEB

Delphi 2007 included a new entrant into the VCL family – VCL for the Web. Based in the IntraWeb technology from AtoZed Software, VCL for the Web allows developers to build web applications (as opposed to web sites) in the same way that standard Delphi clients are built. VCL for the Web is a component-based technology that works very similarly to VCL for Win32. Developers drop components on a form, set their properties, and design web pages just like normal VCL forms. However, when the application is run, it runs in the browser.

VCL for the Web is the first and only tool specifically geared towards creating web applications as opposed to websites. It allows you to build web applications faster and easier than any other tool on the market. Based on a powerful HTML rendering engine, VCL for the Web allows developers to design web applications in the same way as they would normal Windows applications. Using a drag and drop approach, developers can drop controls on forms (which can be thought of as a combination between HTML pages and forms), create events and set properties.

VCL for the Web also makes extensive use of Asynchronous XML and JavaScript technology (AJAX) automatically. Where possible, VCL for the web will automatically inject the necessary AJAX code into the application to take advantage of client-side processing. By simply attaching code to the Async events at design-time, developers can create AJAX based events on the client browser with only Delphi code.

An online demonstration of VCL for the Web can be found on the Embarcadero Developer Network at:

http://windemo1.Embarcadero.com/Tiburon/LaunchReplays/D2009Intraweb/D2009Intraweb.html

DBEXPRESS 4

Database support and component-based access to data have been at the core of Delphi's feature set since the very beginning. Continuing that long premise, Delphi 2007 included a new, underlying data access architecture called dbExpress 4. Completely backwards-compatible with dbExpress 3 at the component level, dbExpress 4 is a complete re-architecture of the VCL's database access layer to provide a single-source, unified means of manipulating data.

dbExpress 4 greatly simplifies the process of building database drivers, thus providing VCL developers with access to a broader range of database servers. Written entirely in Object Pascal, dbExpress 4 provides cross-platform development between native and managed code, complete with connection pooling and command tracing. In addition, it opens up to developers a delegate model that allows them to hook into the data access process and provide additional functionality to the process of reading and writing data. dbExpress 4 also includes significant performance increases over previous versions.

Also new in Delphi 2007 was a rich set of metadata classes that enable developers to both read and write database metadata for any of the nine supported databases.

dbExpress 4 is an underlying architecture, so users of Delphi 2007 saw little difference at the application level. The dbExpress components – the VCL components used to access and expose data to the VCL data-bound controls – have not changed in any significant way. Instead, the code that lies beneath them has been made more efficient and expandable. While the typical user probably will not see much difference, they certainly will recognize the speed enhancements and broader access to data.

Additional Selected Features from RAD Studio 2009

The following section discussed some of the many additional features in RAD Studio 2009 that may be of interest to those reviewing RAD Studio 2010.

The Integrated Development Environment

Code Editor

RAD Studio's code editor is a powerful environment for developing code. It includes numerous features that remove the drudgery of writing code by enabling developers to quickly file the proper method name, write common code constructs using templates, and easily recognize code errors.

Live Templates

Live Templates is a code editor feature that allows developers to quickly and easily write out common code constructs with just a few key strokes. Live templates offer self-describing, intelligent code insertion, and interactive navigation to the variable parts of the template. Live templates allow you to expand small mnemonics into larger code chunks which you can then customize to your needs.

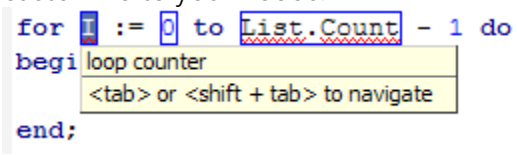


Figure 34 -- A Live Template for constructing a for loop

In Figure above, the developer has invoked the `for` template by simply typing 'for' and pressing the space bar. The template appears, and provides coding entry points for filling out the code template. The developer can move from entry point to entry point using the tab key, filling in the required information as he goes along. The system also provides hints about what is required for each entry point.

Live Templates are simple XML files that describe how the template will work. Thus, developers can easily develop their own Live Templates for specific uses. Live Templates can also take advantage of scripting engines that can perform almost any function at all.

Code Completion

Code Completion is a Code Editor feature that provides the developer with all the information about a class as the developer uses that class, as well as information about identifiers that are declared within the scope of the current code.

Figure 35 -- Code Completion making it easy to complete a complex identifier

Most applications of any significance will contain many identifiers, and often those identifiers will be descriptive and thus sometimes lengthy. Code Completion can make typing such identifiers quick and easy, while at the same time reducing typing errors by ensuring that the correct identifier is entered. As shown in Figure , when Code Completion is invoked, it can easily find the identifier the developer is beginning to type. Simply pressing the `Enter` key will result in the entire identifier being placed in the Code Editor.

In addition, Code Completion can aid in writing code by quickly finding the proper methods and field of a class or record.

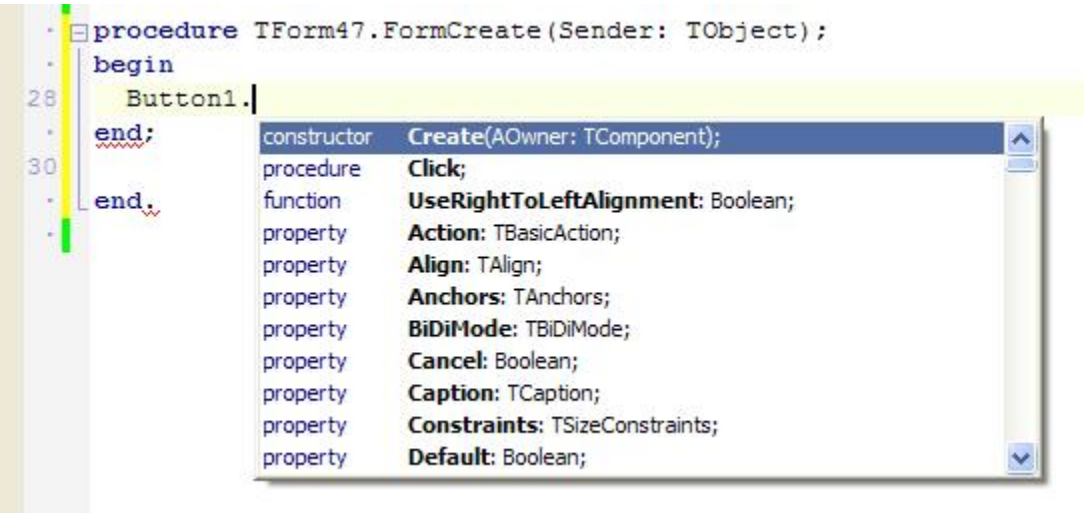


Figure 36 – Code Completion listing all the methods for a TButton variable

In Figure 36, the developer has entered the identifier Button1, added the period, and then invoked Code Completion by pressing `CRTL+SPACE`. As a result, Code Completion has listed all the methods and fields available as part of the `TButton` class. The developer can then either continue typing for the desired method or field, which would result in the list being filtered accordingly, or he can scroll through the list looking for the desired item. Once the correct item is found, a simply `Enter` key will add the selected item in the Code Editor.

Block Completion

Block Completion helps improve the structure of code by ensuring that code blocks are always properly closed. For instance, in Delphi, every `begin` has to have a corresponding `end` statement. All `case` statements also require an `end`. Block Completion ensures that these statements are properly closed without any effort by the developer. If a developer types:

```
begin<enter key is pressed>
```

Block completion will complete the code block by adding the `end` and putting the cursor where the '|' character is, as below:

```
begin
  |
end;
```

Error Insight

Error Insight provides a visual cue to a developer that there is a syntax error in the code editor. Functioning similarly to the "red squiggly lines" of a spell checker, Error Insight can recognize and notify the developer of problems in code.

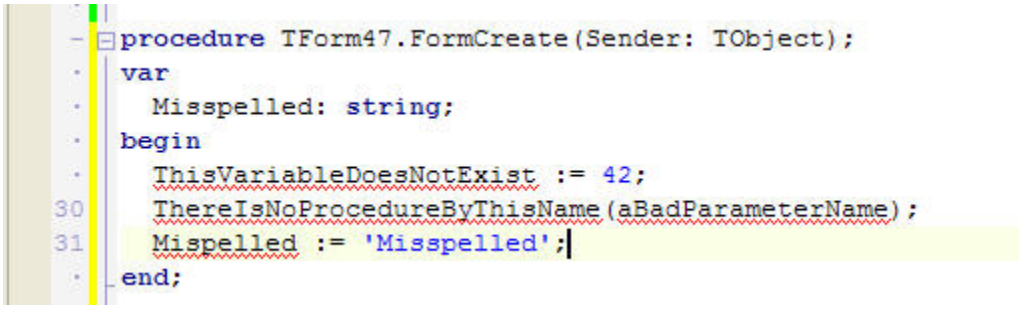


Figure 37 -- Error Insight highlighting code errors

Help Insight

Help Insight provides popup "tool-tips" for identifiers that provide information about the identifier. These tool-tips can be defined in the developer's code using the "triple slash" (///) commenting style with XML tags. Developers can comment their code and have those comments picked up by the IDE and displayed as tooltip help when the mouse cursor is places over the identifier. The style of tool-tips themselves can be formatted by the developer using Cascading Style Sheets.

Figure 38 -- Help Insight, showing the properly formatted comments and the resulting Help Insight tooltip. Note that the text above the class declaration matches the text in the tool-tip window.

THE VISUAL DESIGN EXPERIENCE

FORM DESIGNER

The RAD Studio Form Designer provides the developer with a graphical, event driven development interface. The designer allows for the visual manipulation of components on a form. The developer can drag and drop components from the Tool Palette and lay out a form design, creating a user interface at design-time. Components behave and render very similarly to how they will behave at runtime. Components can be manipulated by the Object Inspector by setting their properties, with the components providing immediate visual feedback to property changes. The designer itself provides valuable visual design aids such as a grid and Visual Guidelines to make the laying out of components of a form a simple and easy process.

Visual Guidelines

Many applications contain forms with numerous controls on them, and as a result, a difficult task facing developers is the alignment of those controls. Often labels need to be properly aligned with the control that they are "labeling". A well designed form will ensure that controls are properly aligned and not strewn about haphazardly.

RAD Studio's Form Designer provides Visual Guidelines that provide indications when controls and text within controls are properly aligned. This allows developers to quickly and easily build forms buy simply aligning controls with the mouse.

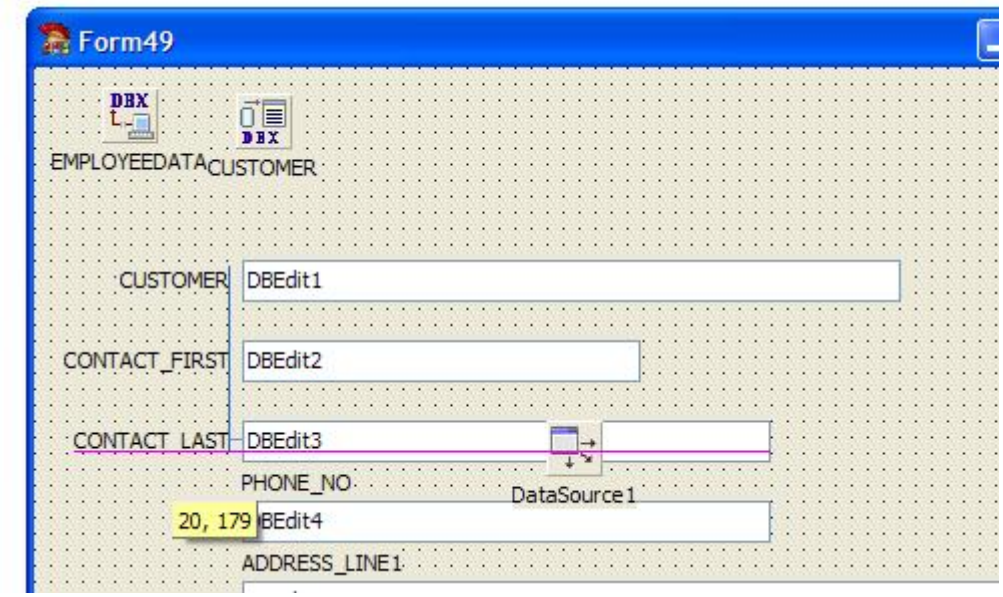


Figure 39 -- Aligning labels and edit boxes with the Visual Guidelines

The Visual Guidelines provide three types of alignment indicators (See Figure above). Blue lines indicate proper alignment of the `Top`, `Bottom`, `Left` and `Right` of components. Magenta indicates the alignment of text elements in controls. And the light grey lines indicate that the controls are spaced properly according to their `Margin` and `Padding` properties.

Object Inspector

The Object Inspector performs two functions: it displays and allows for the setting of a control's properties, and it provides a connection between the events that occur to a component and the code attached to those events.

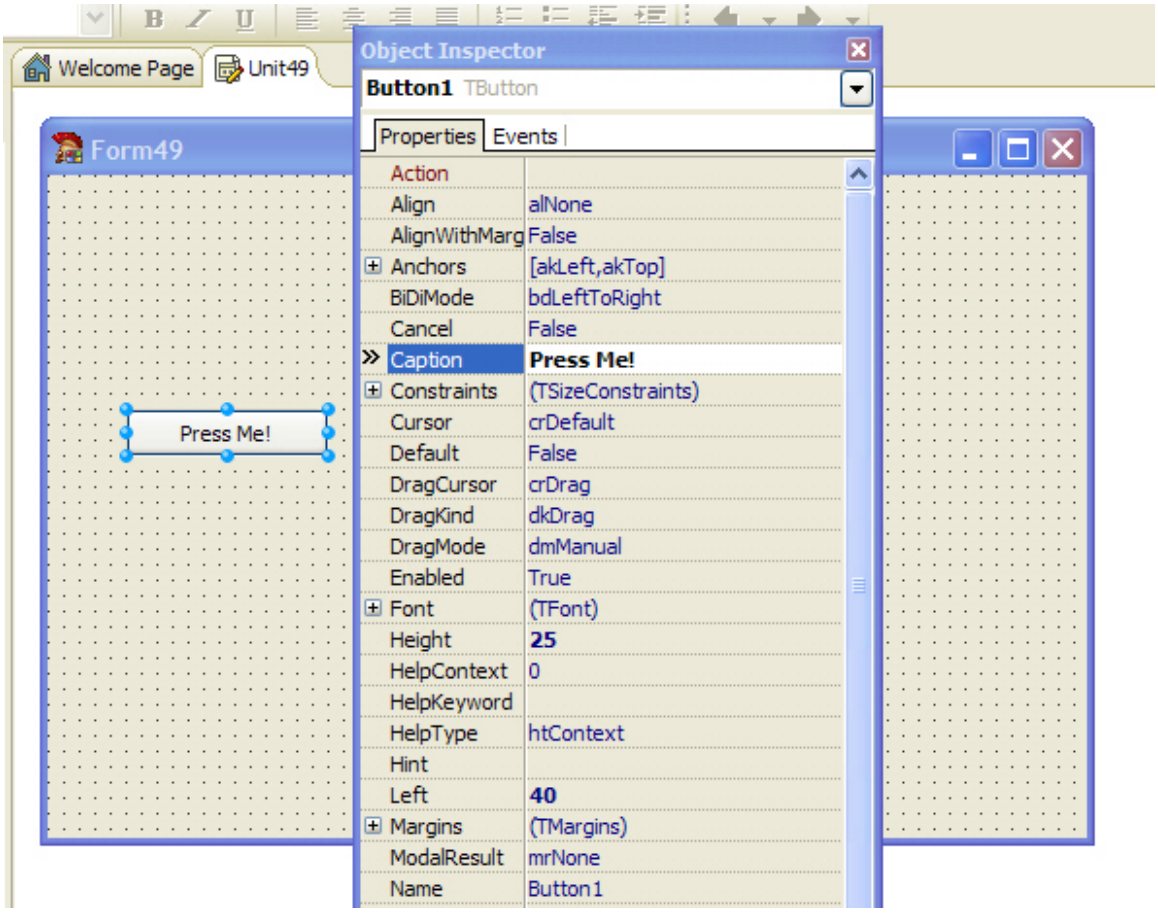


Figure 40 -- The Object Inspector displaying the properties of a TButton component

The Object Inspector is dynamic and automatically displays the properties of any selected control. If more than one control is selected, it displays the properties common to both controls. The Object Inspector indicates properties that have changed from their defaults my making the value **bolded**. Properties can be sorted alphabetically or by category. In addition, the appearance of the Object Inspector can be completely stylized.

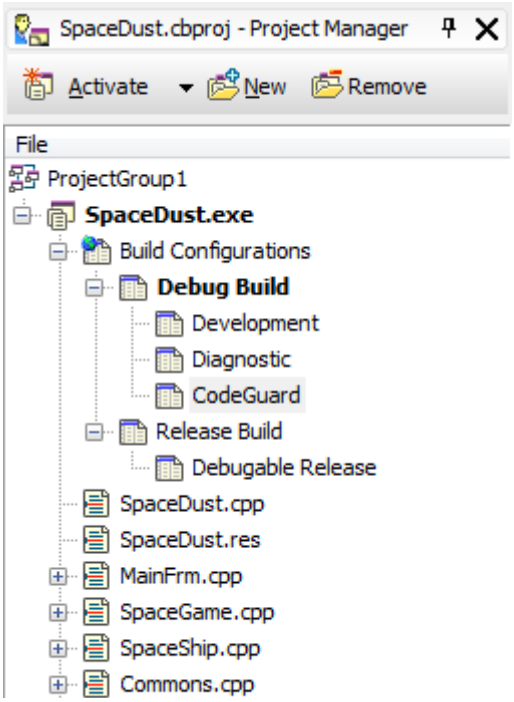
PROJECT MANAGER

The managing of application projects and the code, files, and other resources that go along with them is a key part of any development tool. RAD Studio 2010 provides a powerful Project Manager that enables developers to manage the code files, form files, resources, and other entities that become part of a development project.

The Project Manager provides a hierarchical view of projects. The root is a Project Group which can hold one or more projects. Projects within a Project Group can be rearranged in specific

order, depending up their dependencies. All the projects in a Project Group can be compiled in the order that they are entered into the Project Manager.

BUILD CONFIGURATIONS



Developers can create multiple build configurations for their projects to control their project optimizations, search paths and other options. Build Configurations can be refined, with specialized variants inheriting options from a base configuration.

Build Configurations can be accessed through the Project Manager and all configurations for a project can be build by a single Build command from the context menu.

Projects options can be saved as Option Sets and shared between build configurations in multiple projects at any time.

Figure 41 -- Build Configurations in the Project Manager

The Build Configuration Manager lists all named configurations for all projects in the current project group. It is easy to choose a configuration by name, click 'Select All' and then 'Apply' to make the desired configuration active for all projects.

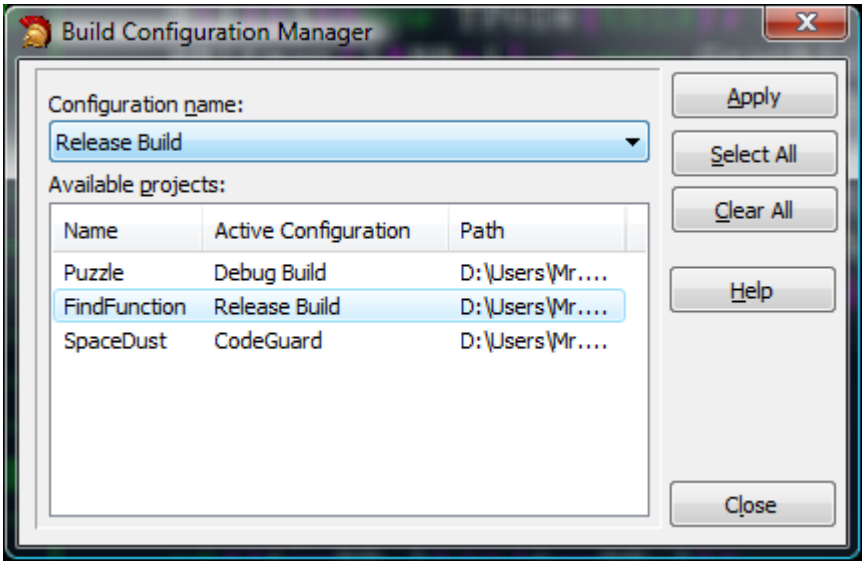


Figure 42 -- The Build Configuration Manager with three projects open

DEBUGGER

RAD Studio includes a complete, full-featured debugger. Delphi's debugger allows the developer complete control over the execution of an application, providing insight all the way down to the machine code being executed. Developers can track the status of any variable, control execution by stepping into or through lines of code. When the debugger is halted on a breakpoint, any variable in scope can be closely and easily examined. The stack trace is available at all times, and generally, debug information for the stack is available, allowing a developer to trace backwards and determine how a particular point or state was arrived at. In short, Delphi's debugger gives the developer total power and freedom to peer into the workings of a running application when searching for execution issues.

Breakpoints

The most common and straightforward of debugging techniques is the use of breakpoints. A breakpoint is a signal that tells the debugger to temporarily suspend execution of your program at a certain point. When the debugger halts code execution at a breakpoint, an application does not terminate or otherwise end the execution of a program. Instead, the debugger suspends execution and allows the developer to peer into the workings and state of the application. Ways in which a developer can view into the state of the application are discussed below.

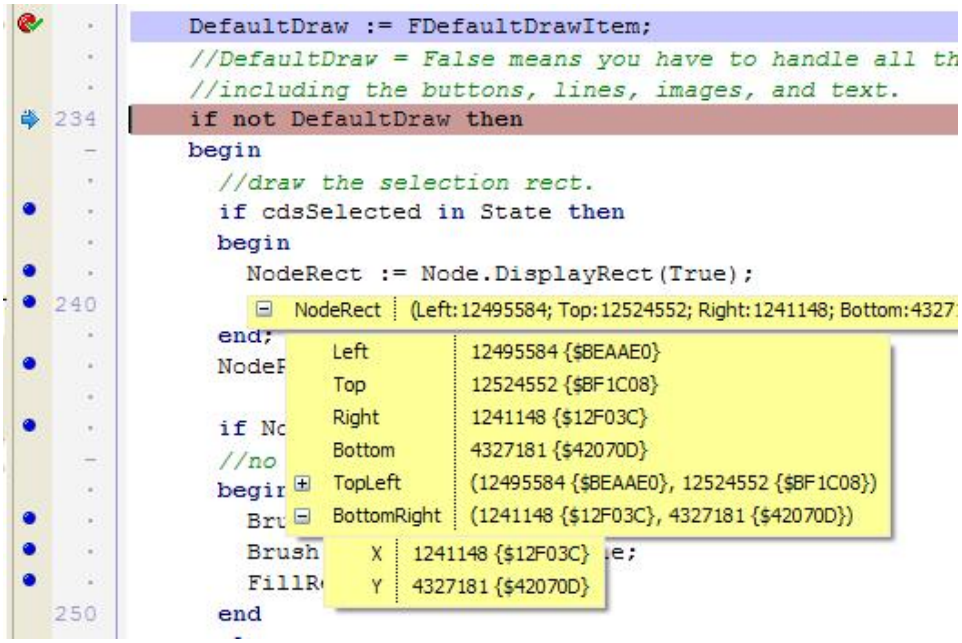


Figure 43 -- The debugger stopped on a breakpoint with an expanded debugger tooltip.

Expandable Watches

Debugger watches are references to specific variables within an application that the developer can use to "keep an eye on" a specific item in code. The debugger will continuously track and report back on the status of any given watch variable. RAD Studio 2010 provides for expandable watches, which all for "drilling down" into the values of complex items that are being watched. Figure above shows a tooltip watch that has been expanded to show the internal information about an object.

CPU View

The CPU View of the debugger gives you a complete view of your application at the machine code level. With it, developers can peer "all the way down to the metal" in an application, including the state of registers, the actual ASM code being executed, the exact binary code of an application, and the state of the FPU. At this level, nothing is left unrevealed. The entire CPU window is shown in Figure ; however, the individual panes of the CPU view can be viewed and docked separately.

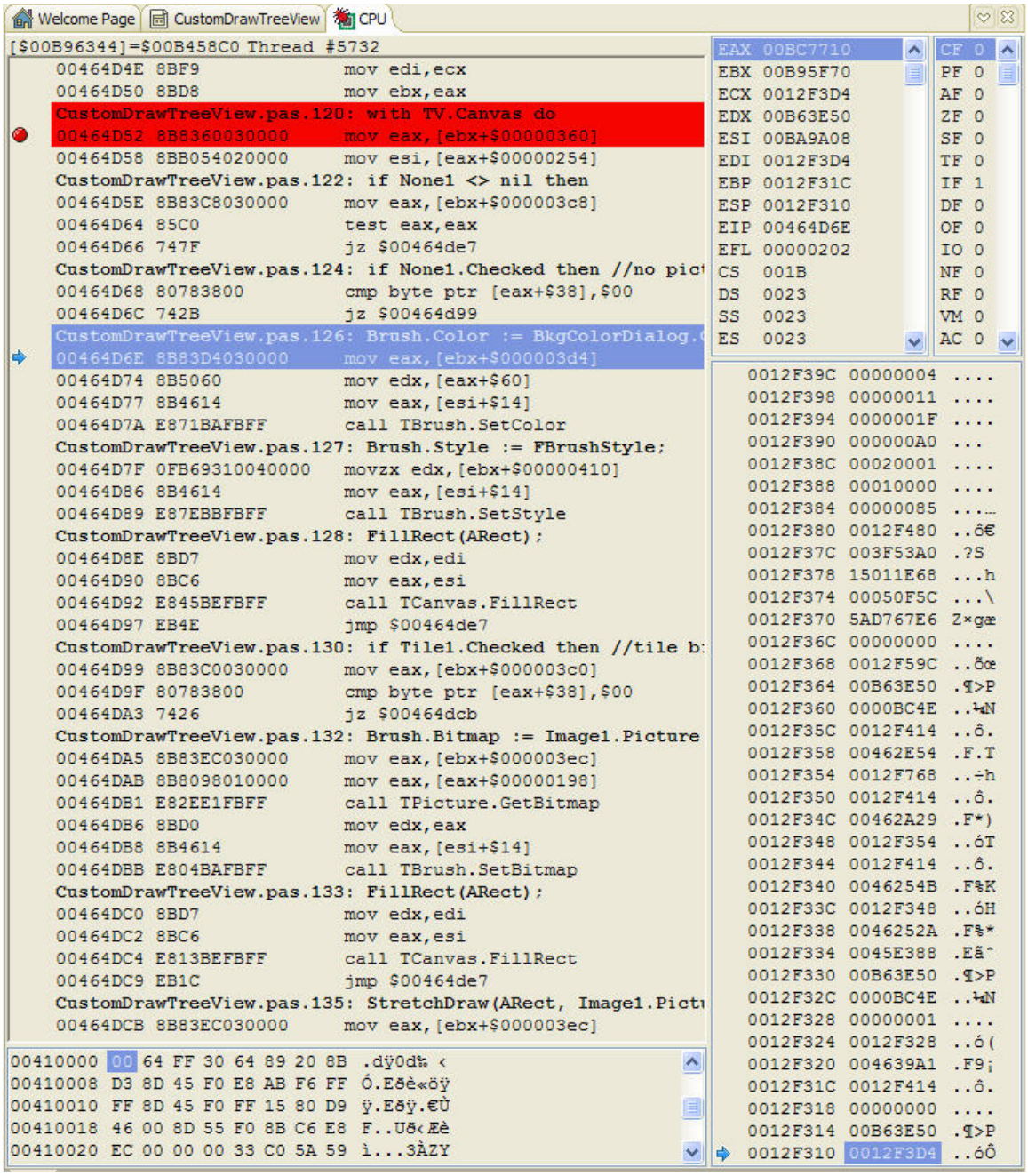


Figure 44 -- The CPU showing a complete, low level view of an application

INTEGRATED UNIT TESTING

Test-driven Development (TDD) has become more and more popular in recent years, and unit testing is a major part of the methodology. RAD Studio 2010 provides support for unit testing by automating the process of writing unit tests for DUnit (DUnit is a Delphi-based library for running unit tests on Delphi code through the VCL language extensions). Developers can designate specific library code for processing by the IDE, and then RAD Studio 2010 will produce a test project and empty test cases for the public and published methods of the selected classes. If and when additional methods are added to a class being tested, the unit test wizard can recognize the new methods and add new stub methods for those new methods. Thus, developers can quickly and easily produce unit tests for code libraries.

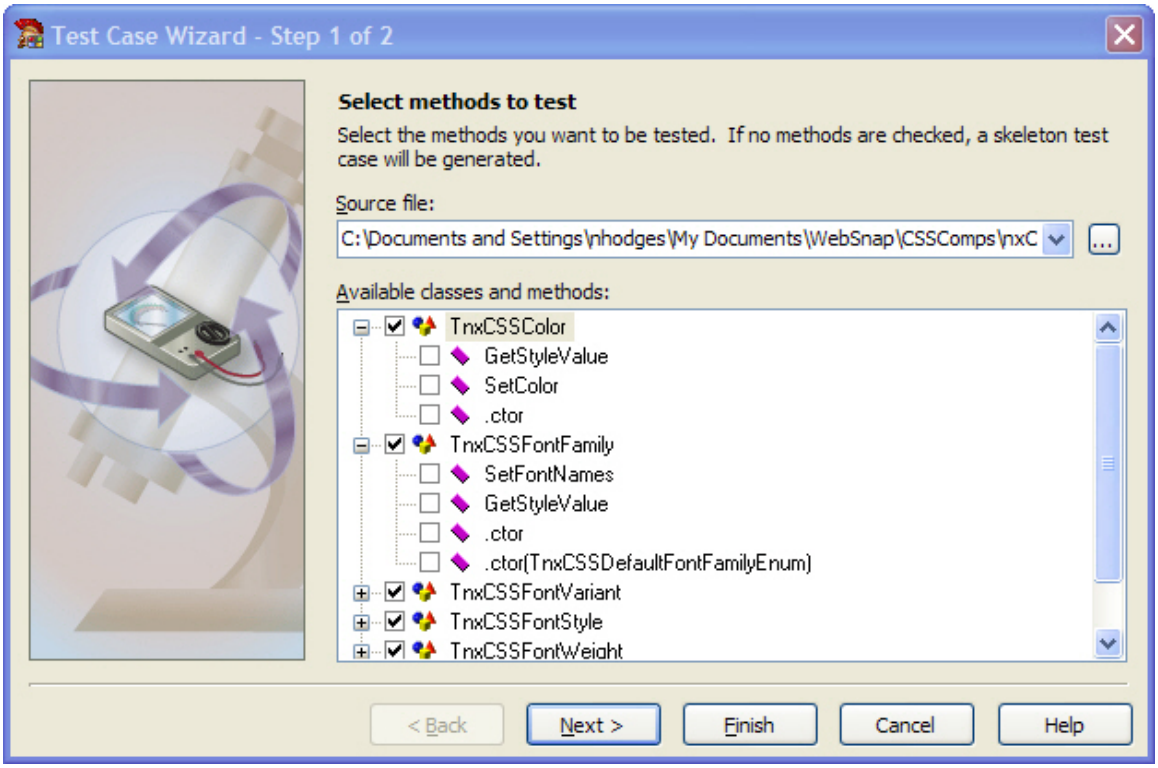


Figure 45 --The Unit Test Wizard gathering information in order to create a set of unit tests for a code library

REFACTORINGS

Refactoring is the restructuring an existing body of code, altering its internal structure without changing its external behavior. The manual refactoring of code has been done since programming was invented, but only in recent years has it been formalized and automated. RAD Studio 2010 provides a rich set of tools to aid developers in refactoring code automatically.

As an example, the Change Parameter refactoring will quickly modify a method declaration and its implementation block. This feature allows for adding and removing parameters, as well as changing the properties of existing parameters.

To use this feature, select a method, function, or procedure in the editor (either its declaration in your class or its implementation) and select `Refactor | Change Params`. Use the Change Parameters dialog box, shown in Figure 46, to make the changes needed.

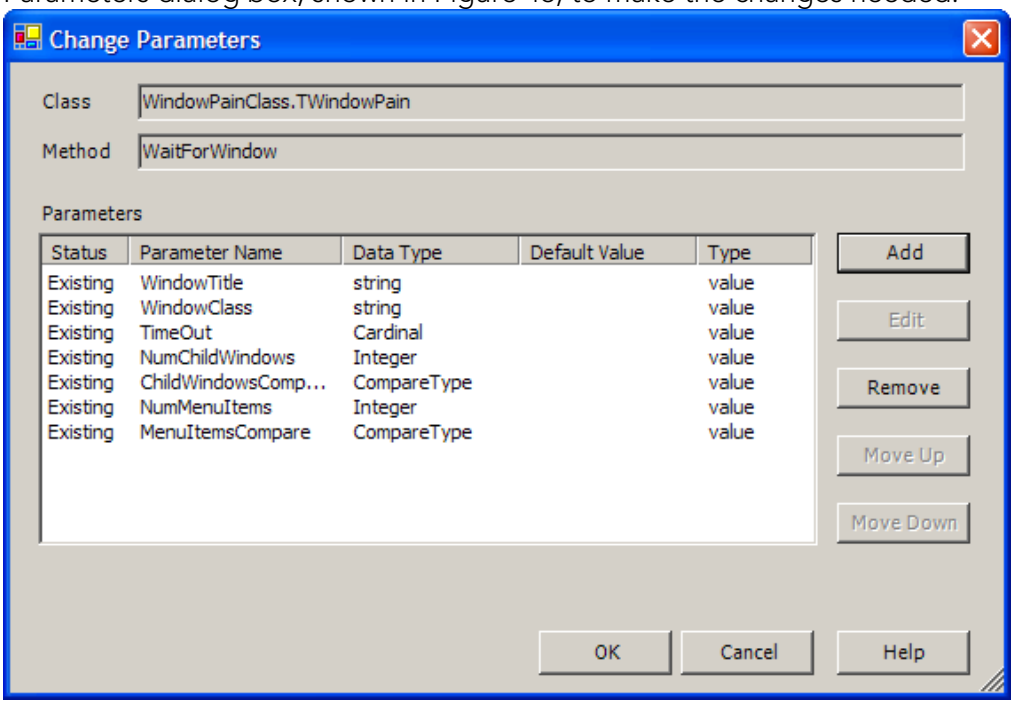


Figure 46 - The Change Parameters dialog box

In addition, a large number of pattern-related refactorings are included in RAD Studio 2010. All the refactorings for Delphi in RAD Studio 2010 are outlined in the table below:

Refactoring Name	Description
Move	Moves static methods from one class to another
Extract Interface	Creates an interface based on selected methods of a class, and declares that interface as implemented by the given class.
Extract Superclass	Moves selected class members to a new class that is the parent of the selected class.
Pull Members Up	Moves selected members to the direct parent class of the selected class
Push Members Down	Moves selected members to the direct descendent class of the selected class.
Safe Delete	Deletes selected item only if it is not used anywhere in the application.
Inline Variable	Replaces instances of a temporary variable with an inline declaration of the literal value.
Introduce Field	Moves an existing local variable to be a field on the given class.

Introduce Variable	Converts a literal expression into a variable declaration with an assignment of the literal expression to the new variable name
Rename	Renames a given identifier throughout the project
Declare Variable	Declares as a local variable that has been used but no yet declared.
Declare Field	For a selected, undeclared identifier, will properly declare said variable as a field on the given class.
Extract Method	Creates a new method based on the selected code, including passing in any necessary parameter.
Extract Resource String	Creates a new resource string based on the selected string, and replaces string literal with name of new resource string.
Change Params	Adds, modifies, and deletes parameters for a given method.

Note that in order to use many of these refactorings, modeling support must be enabled in the project.

THE OBJECT PASCAL LANGUAGE

The Object Pascal language has come a long way from its roots in academia. Far from being the teaching language it originally was, Object Pascal is now a full-featured, object-oriented development language. Object Pascal has kept pace with the latest trends and features in language design. In fact, Object Pascal was the first of the mainstream development languages to implement structured exception handling and properties. Using a highlight readable syntax and a simple, straightforward set of constructs, Delphi is an easy to learn, yet extremely powerful language.

Object Pascal is capable of producing both procedure and object-oriented code. It's a smooth mix of the two coding methods – allowing the developer the best of both worlds. Developers can build and use their own class hierarchy, access and descend from the Visual Component Library (VCL), and create standard procedural code libraries. Delphi has the power to cover the entire range of coding techniques.

Probably the best source of information about The Object Pascal Language can be found online with Marco Cantú's Essential Pascal (in PDF form).

THE VISUAL COMPONENT LIBRARY

The Visual Component Library, or VCL, is a full-featured, eminently expandable and powerful component-based class library. Designed from the ground up to be component-based and thus used in a visual designer, the VCL has performed that role admirably since 1995. Along the way, it has made numerous platform changes, including from Win16 to Win32 and from Win32 to .NET.

DATABASE ACCESS

Delphi provides a simple yet powerful data access model. Based on the VCL class called TDataset, database access is very visual, and very simple to code against. By easily connecting together Connection components to dataset components, tabular data can easily be made available to the VCL's data-bound controls. In fact, data can be displayed without writing any code at all.

In addition, by automatically binding data fields to the VCL's TField descendant classes, developers can easily retrieve and alter data in code with simple to understand code such as this:

```
CustomerTableCustomerNameField.AsString := 'Gary Johnson';
```

In this way, developers can easily access data either visually or in code with little or no effort. The VCL components provide much of the functionality of accessing and managing data without the developer having to write code.

WEB DEVELOPMENT

VCL for the Web

The main tool for building web applications in RAD Studio 2010 is VCL for the Web. Based on the IntraWeb technology from AtoZed Software, VCL for the Web is a component library specifically designed to have a "Delphi-like" feel at design-time, but to produce browser-based web applications at runtime. By very closely mirroring the VCL for Win32, VCL for the Web allows developers that are familiar with the "normal Delphi way" of developing applications to easily transition those skills to building applications that are delivered via the HTML and JavaScript in the browser.

VCL for the Web also provides developer with a very easy, straightforward way to implement AJAX functionality without having to have any knowledge of JavaScript. Developers can execute code on the client by writing Object Pascal code on the server.

MULTI-TIER DEVELOPMENT

The VCL provides a powerful architecture called DataSnap for developing multi-tier architecture. Employing various communications protocols and a powerful set of components, VCL developers can easily and with little effort create an application server that provides data, and a thin client application that can read and write data to and from the application server.

By utilizing a `TRemoteDataModule` class, a developer can easily export an interface to data provider controls that can be remoted using HTTP, DCOM, or a sockets connection. The client application can then obtain a reference to that remote data and import it into a dataset descendent called a `TClientDataset`. Once in a `TClientDataset`, the client application can then perform all the normal "Create, Read, Update, and Delete" (CRUD) operations on the data as the `TClientDataset` tracks and caches all changes. The client application can even be completely disconnected from the server (i.e., the "Briefcase Model"), and then later reconnected. Once reconnected, the `TClientDataset` provides all the code for reconciling the changes made back to the server.

ADDITIONAL MATERIALS

APPLICATIONS BUILT WITH DELPHI

Delphi is a very popular development platform for ISVs because it provides native development with easy deployment requirements. A number of high profile applications are written in Delphi. These include:

- FeedDemon from NewsGator
- The Skype Windows Client
- BeyondCompare from Scooter Software
- The PortableApps Project
- FL Studio Digitial Audio Workstation
- TestComplete and AQTime from AutomatedQA
- FinalBuilder from VSoft Technologies
- InstallAware Install tools
- MediaMonkey Music Management Tool
- Copernic Desktop Search

A more complete list of applications built with Delphi can be found on the public Delphi Wiki.

RAD STUDIO 2010 PRODUCT INFORMATION

Product web pages with data sheets, what's new, edition information, feature matrixes and FAQs

- RAD Studio
- Delphi
- C++Builder
- Delphi Prism

Trial downloads

- RAD Studio
- Delphi
- C++Builder
- Delphi Prism

Thank you for taking the time to evaluate Embarcadero RAD Studio 2010.

We hope you're as excited as we are about the new capabilities of RAD studio 2010, including the ability to build touch/gesture enabled applications, expanded database and DataSnap support, new language features, and enhancements throughout to help developers be more productive, go further, and get their faster with their Windows application development.
.

Embarcadero Technologies, Inc. empowers application developers and database professionals with award-winning tools to design, build and run software applications in the environment they choose. With the acquisition of CodeGear™ from Borland® Software Inc. in 2008, Embarcadero now serves more than three million professionals worldwide with tools that are both interoperable and integrated. From individual software vendors (ISVs) and developers to DBAs, database professionals and large enterprise teams, Embarcadero's tools are used in the most demanding vertical industries in 29 countries and by 90 of the Fortune 100. The company's flagship tools include: Embarcadero® Change Manager™, Embarcadero™ RAD Studio, DBArtisan®, Delphi®, ER/Studio®, JBuilder® and Rapid SQL®. Founded in 1993, Embarcadero is headquartered in San Francisco, with offices located around the world. For more information, visit www.embarcadero.com.