# Table of Contents

Introduction......................................................................................................................... 3  
General Overview of CodeGear RAD Studio 2007........................................................ 3  
Prerequisites................................................................................................................ 3  
Minimum System Requirements................................................................................. 4  
Editions ....................................................................................................................... 4  
  Professional............................................................................................................. 5  
  Enterprise ................................................................................................................ 5  
  Architect.................................................................................................................. 5  
Installation................................................................................................................... 6  
The Integrated Development Environment (IDE) ...................................................... 6  
  General Notes.......................................................................................................... 6  
  Writing Code -- The Code Editor ........................................................................... 7  
  Visually Designing a User Interface -- The Form Designer ................................... 7  
  Managing an Application’s Content -- The Project Manager ................................. 9  
  Looking Under the Hood – The Debugger ........................................................... 10  
  Accessing Data – The Data Explorer .................................................................... 11  
The Visual Component Library ................................................................................ 11  
What’s New in RAD Studio 2007 ................................................................................ 12  
  Blackfish™ SQL....................................................................................................... 13  
  Additional Blackfish SQL Resources ................................................................... 13  
  Support for ASP.NET 2.0 ......................................................................................... 13  
    Additional resources for Delphi and ASP.NET .................................................. 14  
  Vista Support ............................................................................................................ 14  
    Glassing Effects ................................................................................................. 15  
    Vista Dialogs....................................................................................................... 15  
  AJAX and VCL for the Web .................................................................................... 15  
  dbExpress 4 ........................................................................................................... 16  
Selected Features from RAD Studio 2007.................................................................... 16  
The Integrated Development Environment............................................................... 16  
  Code Editor ........................................................................................................... 16  
The Visual Design Experience.................................................................................... 20  
  Form Designer ....................................................................................................... 20  
  Modeling View ...................................................................................................... 22  
  Project Manager .................................................................................................... 22  
  Build Configurations ............................................................................................ 23  
  Debugger ............................................................................................................... 24  
  Integrated Unit Testing .......................................................................................... 26  
  Refactorings .......................................................................................................... 27  
The Object Pascal Language......................................................................................... 29  
The C++ Language..................................................................................................... 29  
The Visual Component Library ................................................................................ 30  
Database Access.......................................................................................................... 30  
Web Development ..................................................................................................... 30  
VCL for the Web ...................................................................................................... 30
Introduction

Thank you for taking the time to review CodeGear RAD Studio 2007. 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 2007, describing its basic purpose and capabilities. This section will be a more high-level overview of RAD Studio 2007’s features. The second part is a summary of the major new features in RAD Studio 2007. This part will discuss the things that are the “latest and greatest” in RAD Studio 2007 – 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 CodeGear RAD Studio 2007

CodeGear RAD Studio 2007 is a general purpose, rapid application development (RAD) Windows application development tool. It is capable of producing both native and managed .NET binaries that execute on x86 operating systems. Native developers can write either Delphi or C++ code, and .NET developers can write code in Delphi for .NET With RAD Studio 2007, developers can build almost any type of Windows binary for both managed and native environments, including stand-alone executables (EXEs) and dynamic link libraries (DLLs), OCX and COM objects, type libraries, Control Panel applets, Windows Service applications, .NET assemblies, and console applications. 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 application servers, web applications, web sites and web services (both clients and servers). RAD Studio 2007 can build web-based applications, full-fledged web sites, thin clients, fat clients, application servers that those clients can access, web services (both clients and servers), windowed applications, ActiveX controls, code libraries accessible by any native or .NET programming language, and multi-threaded applications for running complex embedded systems. In short, RAD Studio 2007 can meet the needs of any developer writing any application for Windows and .NET. It takes a long time and some very serious effort to reach any “virtual programming walls” developing with CodeGear RAD Studio.

Prerequisites

To install RAD Studio 2007, the following prerequisites must be installed:
• Microsoft .NET Framework 2.0
• Microsoft .NET Framework 2.0 Software Development Kit
• Microsoft Direct Access Objects 2.8
• Microsoft Internet Explorer v6.0 SP1
• Microsoft XML Core Services (MSXML) v4.0 SP2
• Microsoft Visual J# .NET v2.0 Redistributable

If you don’t already have the prerequisites installed on your system, the RAD Studio installer will install them for you.

The .NET Framework is required by the IDE, but note that native applications built with Delphi for Win32 or C++Builder have no dependencies at all on the .NET Framework.

**Minimum System Requirements**
The following Windows platforms are supported for installing RAD Studio 2007:

• Microsoft Windows 2000 (SP4 or higher and security updates are recommended)
• Microsoft Windows XP Professional (SP2 or higher)
• Microsoft Windows Vista Professional
• Microsoft Windows Server 2003

The following needs to be installed on the target machine before RAD Studio 2007 can be installed:

• The latest service packs and security updates are recommended for all platforms, including both XP and Server 2003.
• Microsoft Internet Explorer 6.0 SP1 or later
• Approximately 1.5 GB of disk space is required for full installation. The installer requires additional temporary space.
• Pentium II 450MHz or higher (Pentium III 850+ MHz recommended).
• 256 MB RAM (512 MB recommended)
• DVD drive
• VGA or higher resolution monitor
• Mouse or other pointing device

**Note:** The English version of CodeGear RAD Studio 2007 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.

**Editions**
RAD Studio 2007 has three different editions to cater to different market segments – Professional, Enterprise, and Architect.
**Professional**
The Professional edition of RAD Studio 2007 is designed for developers who need a general purpose, RAD Windows development tool with limited or local-only database access. The Professional edition includes the full-featured IDE, implementations of the C++ Language and standard library, Delphi language and Delphi for .NET language, and the complete VCL, including the VCL source code.

The Professional edition contains a limited version of VCL for the Web -- applications are limited to five connections and stand-alone servers.

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

**Enterprise**
The Enterprise edition of RAD Studio 2007 is designed for developers who need access to enterprise level data stored in RDBMSs. The Enterprise edition provides native local and remote access to eight different database engines:

- InterBase 7.5.1 and 2007
- Oracle 10g
- Microsoft SQL Server 2000 and 2005
- DB2 UDB 8.X
- MySQL 4.0.x and 5.0.x
- Informix 9x
- Sybase 12.5
- Adaptive Sybase 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). Reverse engineering of class diagrams (further enhanced with two-way code editing for Delphi developers) gives a high level overview of the whole project. Documentation can be automatically generated providing easy way to explore and review projects, and supporting communication beyond the core team of developers.

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

**Architect**
The Architect edition of RAD Studio 2007 includes everything in the Enterprise version, and adds our advanced Enterprise Core Objects (ECO) IV technology. ECO IV is a
Model-Driven Development (MDD) framework that radically alters and enhances the way that applications are developed. Rather than building databases and code and SQL to support and access those databases, ECO developers can create visual models of the system being developed using UML.

Once the model is created, then ECO covers the task of creating classes that developers can code against for managing all the data and information in the system. In addition, ECO will automatically generate a persistence layer to a wide variety of databases, removing the need for the developer to do anything other than worry about writing code. The real productivity comes when changes need to be made to the system. Changes to the model can be automatically made to both the class structure and persistence layer with the push of a button.

This Reviewer’s Guide will cover ECO more fully below.

**Installation**

RAD Studio 2007 installs either from a DVD or via download from the web. Installation from a DVD is done the traditional way – the customer inserts the DVD into their computer and runs the install application.

However, for the internet download (or Electronic Software Delivery, ESD), the customer receives a small SETUP.EXE stub application that, when run, downloads the necessary binaries from the internet and installs RAD Studio 2007.

The Installation Notes for CodeGear RAD Studio 2007 are found online.

**The Integrated Development Environment (IDE)**

When RAD Studio 2007 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 their 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.
**Writing Code -- The Code Editor**

No mere text editor, RAD Studio 2007’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. 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 their code without introducing errors to make it more readable and organized. CodeInsight produces pop-up windows that give insight into parameters needed for a given routine. ErrorInsight provides immediate feedback by 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.

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

**Visually Designing a User Interface -- The Form Designer**

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 2007’s Form Designer looks exactly like a Window, enabling What-You-See-Is-What-You-Get (WYSIWYG) 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.

![Figure 2 -- The Form Designer using Visual Guidelines to easily align controls on the form.](image)

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 2007. 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.

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 2007’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.

For C++Builder developers, the Project Manager is further enhanced to support multiple build configurations. Option sets can be shared between build configurations. Scriptable events can be run at key points in the build process. Automatically declared variables allow scripts to refer to most properties of the project, such as search path or compiled file name.

**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 2007 provides a debugger integrated into the IDE that provides deep access and insight into an application as it runs.

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.

![Figure 5 -- The Debugger stopped on a breakpoint](image)

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.
**Accessing Data – The Data Explorer**

Many applications require access to data. The IDE includes the DataExplorer – a panel in the ID that provides quick powerful access to database data. In the DataExplorer, a developer can create connections to the RDBMSs supported by dbExpress, Delphi’s database access technology. Once a connection has been created, the DataExplorer 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 DataExplorer to the Form Designer to create data access components on the form.

![Data Explorer](image)

Figure 6 -- The Data Explorer

**The Visual Component Library**

While RAD Studio 2007 can build almost any kind of application, the heart of its main strength lies in building windowed client or stand-alone desktop applications. For this, RAD Studio 2007 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.

In addition to the Win32 version, RAD Studio 2007 also includes VCL.NET. VCL.NET is a code-compatible architecture for building client applications on the .NET managed platform. Applications written in VCL for Win32 can be compiled as managed code applications without any code changes. The discussion of the VCL below applies equally to the native and managed VCL frameworks.
The VCL is the foundation for all Windows application development in RAD Studio 2007. 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 Forms 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 been adapted to support Linux and .NET.

**What's New in RAD Studio 2007**

To a reviewer, the most interesting features in an upgraded product are those that are new to the upgrade. This section will give a look at the features that are new to RAD Studio 2007 since the release of Borland Developer Studio 2006.
Blackfish™ SQL

Included in RAD Studio 2007 is Blackfish SQL RAD Studio 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 the installation of CodeGear RAD Studio 2007, 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 Tognioli, R&D Developer for Blackfish SQL, gave a presentation called Introduction to Blackfish SQL.

Support for ASP.NET 2.0

CodeGear RAD Studio 2007 now provides Delphi developers the ability to build web sites and web applications using the powerful ASP.NET 2.0 framework. RAD Studio 2007 provides full support for master pages, design-time support for all the standard
ASP.NET 2.0 controls, and a complete deployment and debugging solution. RAD Studio 2007 can also import third-party ASP.NET controls.

Additional features included with CodeGear RAD Studio 2007 include:

- Access for any dbExpress database via the DBXDataSource component. ASP.NET is strongly linked to MS SQL Server, and RAD Studio 2007 frees the developer to use the database of their choice.
- A full implementation of the ASP.NET provider model for dbExpress. Rather than being funneled into MS SQL Server, developers can use dbExpress databases including Blackfish SQL and Interbase to store membership, roles, and profile information using the ASP.NET Provider interface. Support for WebParts is included.
- CSS-based layouts. Creating a new ASP.NET application can be daunting for some developers. Staring at a blank webform isn’t always the best way to get started. CodeGear RAD Studio 2007 includes a set of CSS-based layouts that improve the “start-from-scratch-ability” for a project by making the issue of layout and master page setup a breeze. The new Template Libraries feature makes it easy for any developer to create new templates as well.
- CodeGear RAD Studio 2007 also supports developing ASP.NET applications using ECO IV.
- Live templates for easy ASPX coding. CodeGear RAD Studio 2007 includes a set of Live Templates that make coding ASP.NET controls directly in the ASPX file fast and easy.

**Additional resources for Delphi and ASP.NET**

- [Highlander ASP.NET Feature Overview](#)
- [A demonstration of using Master Pages](#) in CodeGear RAD Studio 2007
- [R&D Engineer Jim Tierney demonstrates a number of new features for ASP.NET](#) in CodeGear RAD Studio 2007
- [A demonstration of using Blackfish SQL and ASP.NET](#) together in CodeGear RAD Studio 2007

**Vista Support**

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

Applications built with the Vista support will only function completely as expected on Vista.
**Glassing Effects**

RAD Studio 2007 now provides 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**

RAD Studio 2007 includes 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.
CodeGear RAD Studio 2007 provides VCL for the Web support both and Win32 and ASP.NET.

**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, RAD Studio 2007 includes 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 RAD Studio 2007 is 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 RAD Studio 2007 will see 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.

**Selected Features from RAD Studio 2007**

**The Integrated Development Environment**

**Code Editor**

RAD Studio 2007’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.
In Figure 8 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.

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 9, 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.
In Figure 10, 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:

```delphi
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:

```delphi
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.
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.
The Visual Design Experience

Form Designer
The RAD Studio 2007 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 2007’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 by simply aligning controls with the mouse.

![Figure 13 -- Aligning labels and edit boxes with the Visual Guidelines](image)

The Visual Guidelines provide three types of alignment indicators (See Figure 13 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.

![Object Inspector](image)

**Figure 14 -- 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 by making the value **bolded**. Properties can be sorted alphabetically or by category. In addition, the appearance of the Object Inspector can be completely stylized.
Modeling View

A modeling 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 15 -- UML class diagram generated from a Model View

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 2007 provides a powerful Project Manager that enables Delphi and C++ 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**

C++ 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 16 -- 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 17 -- The Build Configuration Manager with three projects open
Debugger
RAD Studio 2007 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 18 -- 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 2007 provides for expandable watches, which all for “drilling down” into the values of complex items that are being watched. Figure 18 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 19, however, the individual panes of the CPU view can be viewed and docked separately.
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 2007 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, or C++ through the VCL language extensions). Developers can designate specific library code for processing by the IDE, and then RAD Studio 2007 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 20 --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 2007 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 21, to make the changes needed.
In addition, a large number of pattern-related refactorings are included in RAD Studio 2007. All the refactorings for Delphi in RAD Studio 2007 are outlined in the table below:

<table>
<thead>
<tr>
<th>Refactoring Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Move</td>
<td>Moves static methods from one class to another</td>
</tr>
<tr>
<td>Extract Interface</td>
<td>Creates an interface based on selected methods of a class, and declares that interface as implemented by the given class.</td>
</tr>
<tr>
<td>Extract Superclass</td>
<td>Moves selected class members to a new class that is the parent of the selected class.</td>
</tr>
<tr>
<td>Pull Members Up</td>
<td>Moves selected members to the direct parent class of the selected class</td>
</tr>
<tr>
<td>Push Members Down</td>
<td>Moves selected members to the direct descendent class of the selected class</td>
</tr>
<tr>
<td>Safe Delete</td>
<td>Deletes selected item only if it is not used anywhere in the application.</td>
</tr>
<tr>
<td>Inline Variable</td>
<td>Replaces instances of a temporary variable with an inline declaration of the literal value.</td>
</tr>
<tr>
<td>Introduce Field</td>
<td>Moves an existing local variable to be a field on the given class.</td>
</tr>
</tbody>
</table>
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](https://www.marcocantu.com/essential-pascal) (in PDF form).

**The C++ Language**

RAD Studio comes with CodeGear’s implementation of the ISO C++ Language and standard library. C++ is a multi-paradigm language, supporting traditional ‘structured’ code, object oriented development, generic code, generative design and more.
CodeGear’s language extensions to consume Object Pascal code and libraries complements Delphi well, allowing C++ developers full access to component oriented development and the range of 3rd Party Delphi controls.

The implementation in RAD Studio 2007 has been optimized for compile speed, compiling projects up to five times faster than C++Builder 2006!

**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 and C++Builder provide 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:

```pascal
CustomerTable.CustomerNameField.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 2007 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 Material

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
- TestComplete and AQTime from AutomatedQA
- FinalBuilder from VSoft Technologies
- InstallAware Install tools

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

One should note that RAD Studio 2007 is built using itself.

CodeRage I Online Conference Presentations

In March of 2007, CodeGear presented on online conference called CodeRage, in which a number of CodeGear personnel and prominent community members gave demonstrations of the new technology in RAD Studio 2007. Some CodeRage II presentations that may be of interest to reviewers are given below. The links refer to zip files containing Flash presentations of the conference events.

<table>
<thead>
<tr>
<th>Title</th>
<th>Presenter</th>
</tr>
</thead>
</table>

Page 31 of 32
What’s New in Delphi 2007
The New VCL for Delphi 2007
What’s New in DBX4
What’s New in the Delphi 2007 Debugger
Ajax Powered Web Applications with Delphi and IntraWeb 9.0
Web Services in Delphi 2007
MSBuild and the New Project Manager

Nick Hodges, Delphi Product Manager
Seppy Bloom, VCL R&D Engineer
Steve Shaughnessy, Lead R&D Engineer for DBX4
Chris Hesik, R&D Engineer for the Debugger UI
Olaf Monien, AtoZed Software
Bruneau Babet, Delphi R&D Engineer
Dave Wilhelm — R&D Engineer for the IDE

CodeRage II Presentations

CodeRage I was followed up in November 2007 by CodeRage II, an even bigger online event for CodeGear developers. Presentations given in that event cover features and uses of RAD Studio are listed below. (Note that these files are ZIP files that contain a Flash version of the presentation embedded in an HTML file.)

<table>
<thead>
<tr>
<th>Title</th>
<th>Presenter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delphi Product Address</td>
<td>Nick Hodges – Delphi Product Manager</td>
</tr>
<tr>
<td>Top Ten Things Added to Delphi since Delphi 7</td>
<td>Pawel Glowacki – Delphi EMEA Evangelist</td>
</tr>
<tr>
<td>Introduction to Blackfish SQL</td>
<td>Leonel Tognioli – CodeGear R&amp;D Developer</td>
</tr>
<tr>
<td>Delphi 2007 and Vista</td>
<td>Marco Cantu – WinTech Italia</td>
</tr>
<tr>
<td>Delphi Dynamic Architectures with Packages</td>
<td>Marco Cantu</td>
</tr>
<tr>
<td>Introduction to Generics in Delphi</td>
<td>Ray Konopka – Raize Software</td>
</tr>
<tr>
<td>Creating Composite VCL Controls</td>
<td>Ray Konopka</td>
</tr>
<tr>
<td>Building Advanced VCL.NET ECO Applications and Migrating Old Databases</td>
<td>Holger Flick – Delphi Q&amp;A Engineer</td>
</tr>
<tr>
<td>Delphi Unit Testing Power Tools</td>
<td>Craig Stuntz — Vertex Systems Corporation</td>
</tr>
<tr>
<td>Effective UML Modeling in Delphi</td>
<td>Alexander Evdogimov and Alexey Dubkov — CodeGear</td>
</tr>
<tr>
<td>ClientDataset Tips and Tricks</td>
<td>Cary Jensen – Jensen Data Systems</td>
</tr>
</tbody>
</table>