

# Delphi 2007 for Win32

## The first personality of the CodeGear RAD Studio

by Bob Swart

This time, I will examine Delphi 2007 for Win32, the first personality of the CodeGear RAD Studio, which was released in the first quarter of 2007. I've upgraded to Delphi 2007 for Win32 myself, and will report next time in more detail on VCL for the Web and DBX4 experiences in real projects.

### Delphi 2007

Back in February of this year, CodeGear announced "Delphi 2007 for Win32". The name specifically mentions Win32, holding the implicit promise for more/other Delphi 2007 editions to come (like a Delphi 2007 for C++ as well as a Delphi 2007 for .NET 2.0 - both to be expected later this year).

During the closing session of CodeRage - the first "virtual" online conference organised by CodeGear - on March 19th, the immediate availability of Delphi 2007 for Win32 was announced. Customers could purchase it and either wait for the DVD or download a small setup.exe to install it immediately.



Of course, the number 7 has a special meaning for many people (often even more special than 42, although that number has a deeper meaning). I still have fond memories of Turbo and Borland Pascal version 7, and Delphi 7 doesn't get the chance to collect a lot of magnetic dust on my machine. In fact, Delphi 7 appears to be among the most popular versions of Delphi being used today (for Win32 development, that is). Delphi 2007 for Win32 is supposed to seduce the Delphi 7 developers to stepping up to the latest edition.

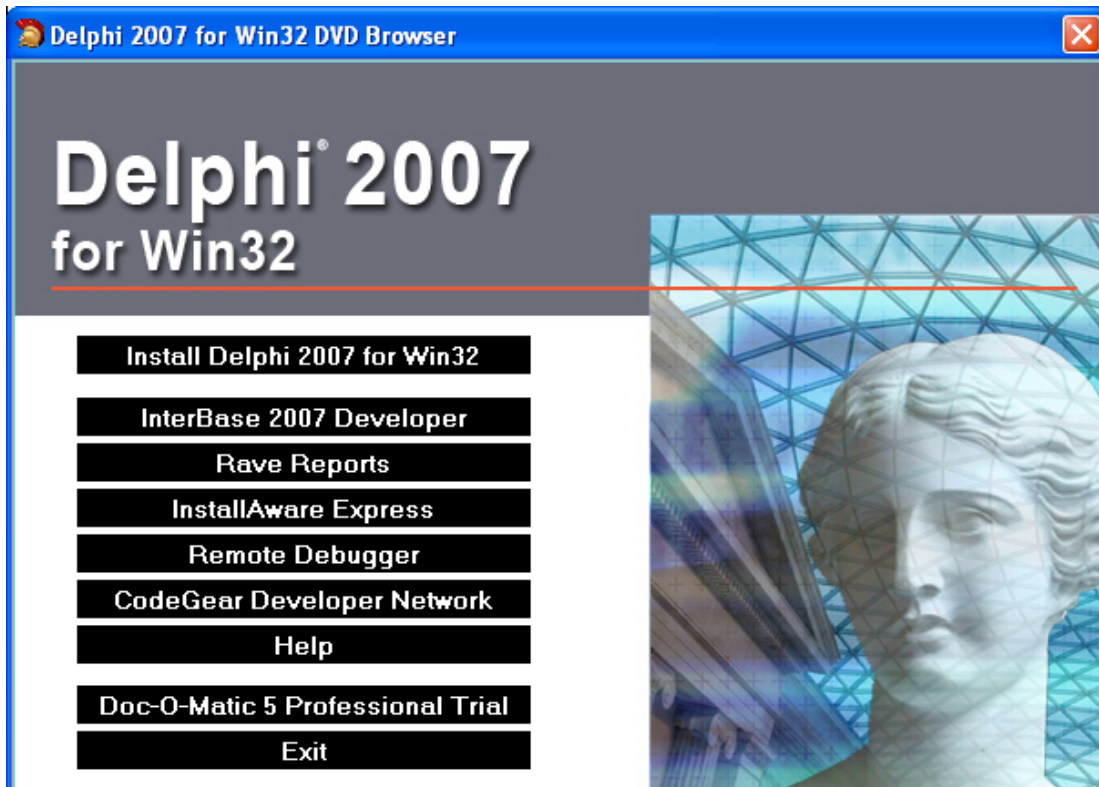
### CodeGear RAD Studio

Although I've been talking about Delphi 2007 for Win32 so far, the actual title of the application is CodeGear RAD Studio. And Delphi 2007 for Win32 is just the first part of the entire RAD Studio, which will also consist of a Delphi 2007 for C++ and Delphi 2007 for .NET (both yet to come). These later editions of Delphi 2007 should integrate seamlessly in the CodeGear RAD Studio. In fact, from what I've been told, if you purchase Delphi 2007 for Win32 now, you actually purchase a license for the CodeGear RAD Studio and will get the forthcoming Delphi 2007 editions as well (as they become available). So even if you're mainly waiting for .NET 2.0 support, it's no use waiting until Delphi 2007 for .NET 2.0 is released, because it'll be part of the same CodeGear RAD Studio. Again, this is what I've understood, but make sure to check with your local CodeGear representative when you purchase Delphi 2007 for Win32 just in case...

One thing is certain: Delphi 2007 for Win32 is not the same as a (unreleased) Turbo Delphi 2007 for Win32. The latter is a single-personality IDE, available in Personal and Professional editions only, while Delphi 2007 for Win32 is - as stated before - part of the full CodeGear RAD Studio. Furthermore, the Turbo editions of Delphi 2007 will not be released until all personalities of the CodeGear RAD Studio have been released and integrated (just like the Turbo Delphi 2006 editions were released several months after the full Borland Developer Studio 2006 was available).

## Installation

During the installation of Delphi 2007 for Win32 (the first personality of the CodeGear RAD Studio), you may wonder why the .NET Framework version 2.0 is needed. This is of course due to the fact that the CodeGear RAD Studio itself needs .NET 2.0. Not to produce .NET code, but for the IDE itself (which regards the .NET Framework, apart from a development framework, also as a useful library to use).



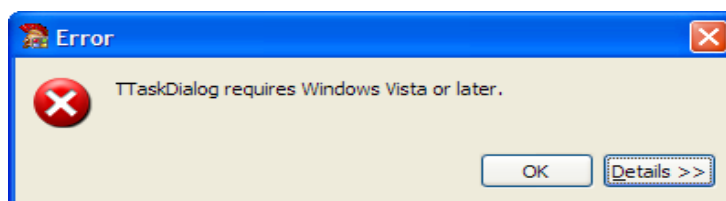
The installation process - handled by the InstallAware Wizard - checks all prerequisites which are either taken from the DVD or downloaded from the internet when needed. The DVD contains almost the full available 4 GB of data - all different editions as well as languages of Delphi 2007 for Win32 are present on it (based on the Product License Key, the right version will be installed).

And apart from all Delphi 2007 for Win32 versions and editions (plus their prerequisites), the DVD also contains InterBase 2007 Developer edition, Rave Reports, the InstallAware Express version, the Remote Debugger, as well as a copy of the CodeGear Developer Network and finally a trial version of Doc-O-Matic Professional.

## Windows Vista Support

One of the big selling points of Delphi 2007 for Win32 is not the fact that it's for Win32, but mainly that it contains support for Windows Vista - for Win32 applications.

The Windows Vista support can be found in a number of explicit places, more notably the Vista Dialogs components TFileOpenDialog, TFileSaveDialog and TTaskDialog. These three components in the Vista Dialogs category represent three new Windows Vista dialogs. However, unless you know beforehand that your client application will run exclusively on Windows Vista, I would recommend against using them. At least the former two. The reason is that these three new dialogs actually use the dialogs within Windows Vista, and will fail with an EPlatformVersionException when you run your application on another version of Windows:



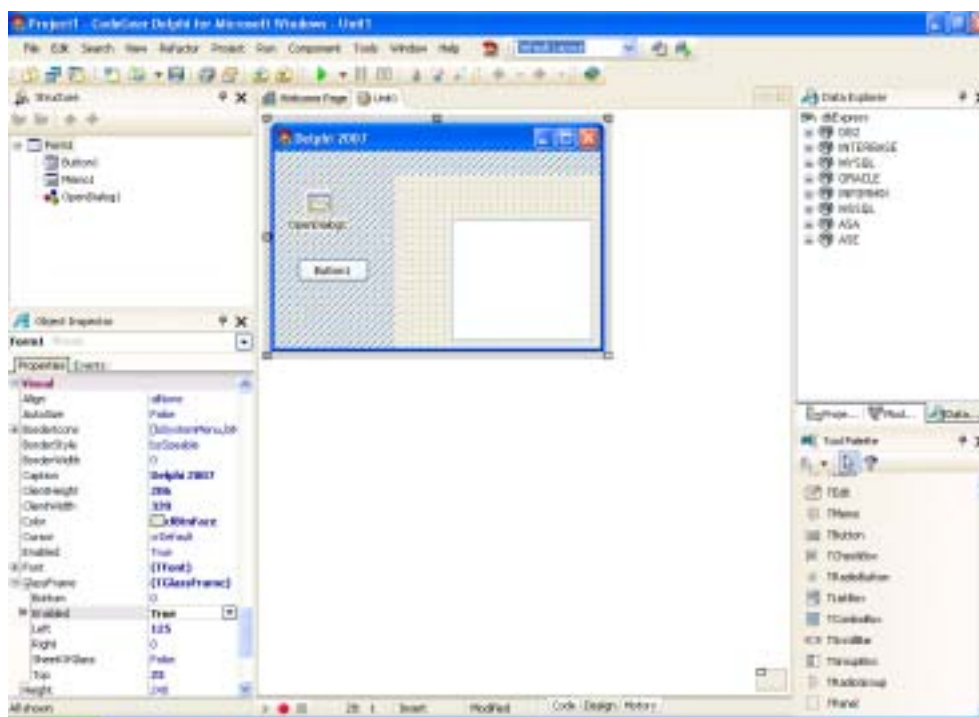
Instead of using the explicit Windows Vista editions TFileOpenDialog and TFileSaveDialog (which can result in the exception when not run on Windows Vista), we can also use the original TOpenDialog and TSaveDialog, and set the UseLatestCommonDialogs global variable (found in the Dialogs unit) to True. This will also enforce that the latest common dialogs are used, resulting in the Windows Vista open and save file dialogs when running on Windows Vista, but the “normal” open and save dialogs when running on previous editions of Windows. The best of both worlds, and even better since you do not even have to explicitly set UseLatestCommonDialogs to True (it’s set to True by default). You only have to recompile your existing application with Delphi 2007 for Win32 to enable the Windows Vista file open and file save dialogs!

Apart from the three new Windows Vista specific dialogs, a number of Windows Vista specific support classes are also available in Delphi 2007 now, including (according to the list in the online help): TCustomFileDialog, TCustomFileOpenDialog, TCustomFileSaveDialog, TCustomTaskDialog, TFavoriteLinkItem, TFavoriteLinkItems, TFavoriteLinkItemsEnumerator, TFileTypeItem, TFileTypeItems, TTaskDialogBaseButtonItem, TTaskDialogButtonItem, TTaskDialogButtons, TTaskDialogButtonsEnumerator, TTaskDialogProgressBar, and TTaskDialogRadioButtonItem.

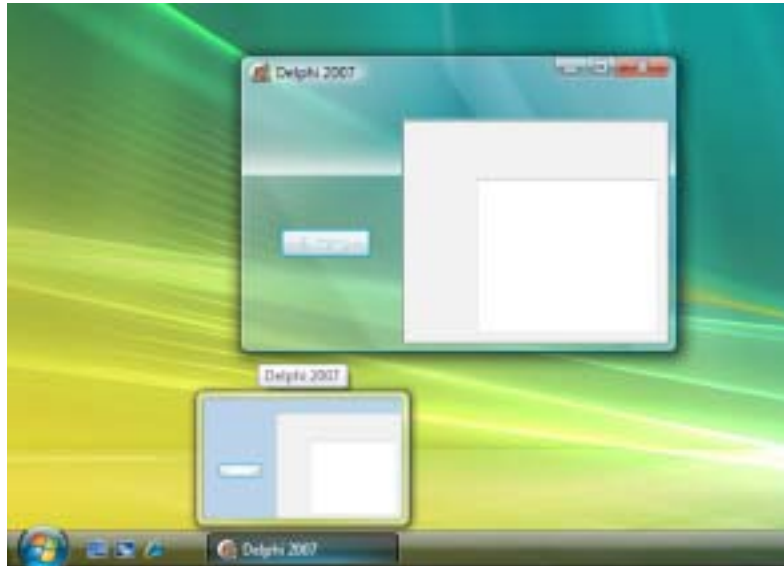
## Glass Frame

Another Vista specific new feature in Delphi 2007 has been added to the TForm, which now has a GlassFrame property in the Visual Category. This GlassFrame property controls the behaviour of the titlebar of the application, and optionally all the way down to cover the entire form. This time, in order to enable the effect, you need to set the Enabled subproperty of the GlassFrame property to True (unlike the UseLatestCommonDialogs, the GlassFrame is not enabled by default). With the GlassFrame.Enabled set to True, the titlebar will show the glass effect when running on Windows Vista with Aero enabled as well (this of course depends on your configuration of Windows Vista – for one of my machines I’m still waiting for an updated Radeon 9250 driver for Windows Vista in order to be able to use Aero).

Apart from the Enabled subproperty, the GlassFrame property includes some additional properties for extra effect. You can set the Bottom, Left, Right and Top property to specify which borders of the Form will be turned into glass as well. This can give an extra nice effect (again, only on Windows Vista with Aero enabled). When designing the Form, the effect can be previewed using blue diagonal lines, as can be seen in the screenshot below (which also shows the GlassFrame property in the Object Inspector).



The following screenshot shows the effect of my GlassFrame settings with the Enabled property set to True, the Left to 125 and the Top to 25 pixels.



Note that you can also turn the entire Form into glass by setting the `SheetOfGlass` property to `True`. Although nice, I must say that I've found no practical application for that particular option so far.

Although the `GlassFrame` only works on Windows Vista with Aero enabled, there has been an (almost) similar feature in Delphi 7 supported by the `AlphaBlend` and `AlphaBlendValue` properties. By setting `AlphaBlend` to `True`, we can control the opaqueness of the Form with the `AlphaBlendValue` going from 0 (completely invisible) to 255 (totally visible). I've been using the `AlphaBlendValue` to have special forms like the About box or splash screen fade in and out. But apart from that, it's not much more than a cosmetic gimmick, if you ask me.

One other notable enhancement found in Delphi 2007 compared to previous versions of Delphi is that (XP) Theme support is now built-in from the start, meaning that you no longer have to add the explicit `XPManifest` component on a form (which in itself was not needed, but was only used to add the `{SR WindowsXP.res}` resources to your executable). All applications compiled with Delphi 2007 will have theme support enabled from the start. If you want to disable the themes for your application, you need to uncheck the "Enable runtime themes" checkbox in the Application page of the Project Options. This will affect the project's `.res` file, as well as the executable itself after the next recompile.

If you want to control the behaviour without recompiling your application, you first need to uncheck the "Enable runtime themes" option, and then you can use the presence of a `..exe.manifest` file next to your `.exe` with the following contents:

---

```
<dependency>
  <dependentAssembly>
    <assemblyIdentity
      type="win32"
      name="Microsoft.Windows.Common-Controls"
      version="6.0.0.0"
      publicKeyToken="6595b64144ccf1df"
      language="*"
      processorArchitecture="*" />
    </dependentAssembly>
  </dependency>
```

---

When present, this manifest will ensure that the corresponding `.exe` will be using themes. Without the manifest file present, the executable will not use themes. If you already have a `.exe.manifest` file, then just make sure the specific entry is present or not depending on your theme wishes.

The same technique can be used to disable themes for the Delphi 2007 for Win32 IDE itself. For this, you need to edit the `bds.exe.manifest` file in the `C:\Program Files\CodeGear\RAD Studio\5.0\Bin` directory. By default, the `<dependency>` section is present in the `bds.exe.manifest` file, but you can remove it if you wish to disable theme support in the IDE (not for compiled applications, but only for the IDE itself).

## VCL for the Web and AJAX

Perhaps a bit more relevant to Web developers is the AJAX support found in Delphi 2007. The press release even mentioned VCL for the Web. Unfortunately (but with all due respect), this is not something completely new, and like AJAX can be seen as a new name for something we already knew (and sometimes have been using for a while). VCL for the Web is in fact the (new?) name for the latest (special?) Delphi VCL version of IntraWeb. And IntraWeb has been extended with AJAX support.

IntraWeb, a third-party product originally known by the not well-marketable name Portcullis, offers true RAD development for the web, and was first added (version 5.x) to Delphi 7. Delphi 2007 now includes IntraWeb version 9.0.14. Like the original bundling with Delphi 7, there are two different editions: IntraWeb in Page mode and IntraWeb in Application mode, the later only fully available to users of the Enterprise edition of Delphi (or only in trial-mode and available as a separate purchase from AtoZedSoftware).

Since IntraWeb offers VCL components that can be used to build web editions of VCL GUI applications with similar controls (edits, buttons, listboxes, grids, etc.) the learning curve for building RAD web applications is low, and the name VCL for the Web is something I can agree with 100%. You only have to realise that it's not something new (only the name is).

The AJAX support in IntraWeb will be covered in more detail next time. I can tell you that VCL for the Web makes it a lot easier to add AJAX capabilities to your web applications, without having to know all kinds of HTML, JavaScript or CSS details.

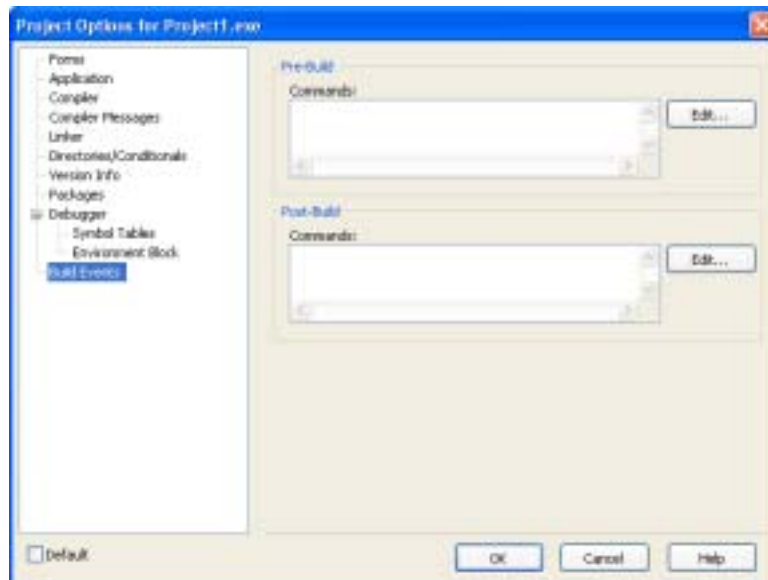
## DBX4

Another enhancement found in Delphi 2007, which may not be immediately obvious to all users, is DBX4. Capable of using older dbExpress 3 drivers (so your applications written in Delphi 2006 will still work), but not dbExpress 2.5 drivers (so applications written in Delphi 6 or 7 will need some work), DBX4 is a rewrite of the cross-platform data access layer. Cross platform, since it now supports Win32 as well as .NET (no longer Linux), supports connection pooling, and also makes it easier to write your own drivers. Next time, I'll cover DBX4 in some more detail.

Delphi 2007 Professional contains DBX4 drivers for InterBase and MySQL version 4.1 and 5.x. The Enterprise edition of Delphi 2007 adds Oracle 10g, MS SQL Server 2000 and 2005, Informix 9.x, IBM DB2 UDB 8.x and Sybase 12.5 to this list.

## MSBuild

Delphi 2007 uses MSBuild now, which also results in the ability to have pre-build and post-build commands (as can be seen in the Build Events project options dialog).



Finally, specifically for Delphi 7 developers, there are a number of articles that explain the new IDE features since Delphi 7 (at <http://dn.codegear.com/article/34323>), the new VCL features since Delphi 7 (at <http://dn.codegear.com/article/34325>), the new Delphi language features since Delphi 7 (at <http://dn.codegear.com/article/34324>), and the new Debugger enhancement since Delphi 7 (at <http://blogs.codegear.com/chrisshesik/archive/2007/04/13/33946.aspx>). Recommended reading!