

CodeGear™ Delphi® for PHP 2.0

Reviewer's Guide



The RAD Visual PHP Development Environment

Table of Contents

Table of Contents.....	2
Introduction.....	3
General Overview of Delphi for PHP 2.0.....	3
Minimum System Requirements.....	3
Installation.....	3
What's New in Delphi for PHP 2.0	4
Developing with Delphi for PHP 2.0.....	4
Enhancements to VCL for PHP	6
Enhancement to the Delphi for PHP 2.0 IDE and Code Editor	7
Other Delphi for PHP 2.0 Enhancements	10
Delphi for PHP 2.0 in Depth.....	11
The Delphi for PHP 2.0 Integrated Development Environment.....	11
The Visual Component Library (VCL) for PHP 2.0.....	15
Additional Materials	18
The Delphi for PHP 2.0 Documentation Wiki.....	18

Introduction

Thank you for taking the time to review Delphi for PHP 2.0. 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 Delphi for PHP 2.0, describing its basic purpose and capabilities. This section will be a more high-level overview Delphi for PHP 2.0's features. The second part is a summary of the major new features in Delphi for PHP 2.0. This part will discuss the things that are the "latest and greatest" in Delphi for PHP 2.0 – 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 Delphi for PHP 2.0

Delphi for PHP 2.0 is a rapid application development based integrated development environment (IDE) for developing PHP applications. Delphi for PHP 2.0 includes a powerful visual framework, the Visual Component Library for PHP (VCL for PHP), that provides developers with a drag-n-drop development environment. The VCL for PHP provides the huge productivity gains for PHP developers that Windows developers have come to expect using tools like Delphi and C++Builder. At the same time, Delphi for PHP 2.0 provides PHP developers with a powerful and productive editing environment that PHP developers who like to "hand-tweak" their PHP code have come to expect. One of the main themes of the Delphi for PHP 2.0 release is providing productivity both via a rapid application development environment and via a product code editor.

Minimum System Requirements

Delphi for PHP 2.0 has the following system requirements:

- 512 MB RAM (1 GB for Microsoft® Windows Vista™)
- 200 MB free hard disk space
- CD-ROM or DVD-ROM drive
- SVGA or higher resolution monitor (XGA recommended)
- Intel® Pentium® III/M 1.4 GHz, or Pentium IV 1.4 GHz minimum
- Delphi for PHP 2.0 will run on the following Windows versions:
 - Microsoft® Windows Vista™
 - Microsoft® Windows Server® 2003 (SP1)
 - Microsoft® XP Professional (SP2)
 - Windows® 2000 Professional (SP4)
 - Windows® 2000 Server (SP4)

Installation

Delphi for PHP 2.0 installs either from a CD-ROM or via download from the web. Both types of installation are done in the traditional way – the customer runs a standard install executable that gathers information from the user and then installs the application.

What's New in Delphi for PHP 2.0

Delphi for PHP 2.0 contains many new features that increase the capabilities and productivity of PHP developers. This release focuses on several key areas:

1. An expanded and improved visual development experience that goes beyond just PHP with a new visual HTML editor and templated forms designer to enable the developer to work with both HTML and PHP to build complete web pages rather than switching between PHP and HTML design tools.
2. Increased focus on making it easier to develop data-driven applications with simplified database connectivity to additional databases, easier drag-and-drop development of database apps, and new Oracle database components.
3. Enhancements to the VCL for PHP to increase the capabilities and productivity of doing RAD design in PHP. Delphi for PHP 2.0's main strength lies in the tremendous productivity gains PHP developers can get from doing visual web form design using rapid application development techniques. The VCL for PHP is a powerful, feature rich, and extensible framework that enables visual design and development of PHP applications. Delphi for PHP 2.0 has a many enhancements that further empower PHP developers/
4. Enhancements to the code editor to speed the development of code and to enable PHP developers to rapidly write their own PHP code in an efficient and productive manner. PHP developers are first and foremost writers of code, and as such, they require a powerful, full-featured editor. Delphi for PHP 2.0 has placed a heavy emphasis on improving the PHP developer's coding efficiency.

Developing with Delphi for PHP 2.0

Developing PHP applications with Delphi for PHP 2.0 is different than the traditional means of using PHP. Traditionally, PHP development has been editor and text intensive – developers have written PHP websites and applications in powerful editors, with little help from visual design tools. Delphi for PHP 2.0 changes all that.

By providing a rapid application, drag-and-drop means of developing applications, Delphi for PHP 2.0 developers now have the ability to drastically increase their productivity while at the same time continuing to produce pure PHP/HTML-based solutions. Traditional Windows development tools like Delphi have long provided developers with RAD development, and now Delphi for PHP 2.0 brings this powerful advantage to PHP.

For instance, imagine a very simple PHP application that simply takes user input from a text box and puts it into a list box. Doing such a PHP application might take an hour or more using traditional development techniques of hand-coding all the necessary PHP. Displaying data could take even longer.

Develop Faster with Delphi for PHP 2.0

But with Delphi for PHP 2.0, it is literally a matter of one minute or less of development time to produce such an application. For instance, to produce the simple application described above, merely follow these simple steps:

1. Select File|New Application from the main menu.
2. Go to the Tool Palette and click the button component, and then click again on the form designer. A Button should be placed on the form.
3. Do the same for the ListBox and Edit components.
4. Double click on the button. The IDE will create an event handler called Button1Click that will execute when the application is run and the button is pressed. In that event handler, add the following code:

```
$this->ListBox1->addItem($this->Edit1->Text);
```

5. Save the application to any desired directory
6. Press the Run button (or press F9)
7. Once the application is running, type some text into the edit box, and then press the button. The text entered should immediately appear in the listbox.

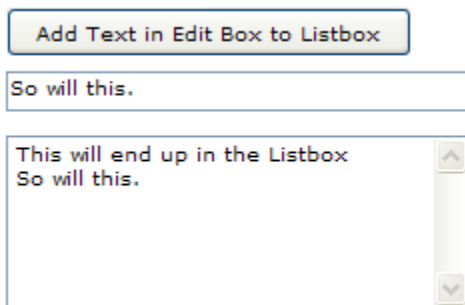


Figure 1 -- A simple PHP application running in a browser and using three components and one line of developer written code.

By leveraging the power of the VCL for PHP, tasks that traditionally have required a large amount of coding can be reduced to a few mouse clicks.

Accessing Data with Delphi for PHP 2.0

Accessing data can be done in much the same way. To illustrate, take the following simple steps in Delphi for PHP 2.0:

1. Ensure that the InterBase Developer Edition Server (which is installed as an option with Delphi for PHP 2.0) is running.
2. Select File|New Application from the main menu.
3. Go to the Data Explorer and right click on the InterBase node and select "Register Database".
4. In the resulting dialog, set the following properties:
 - a. **Connection Name:** Test

- b. **Database Name:** C:\Program Files\Common Files\CodeGear Shared\Data\EMPLOYEE.GDB
 - c. **User Name:** SYSDBA
 - d. **Password:** masterkey
5. Note that once this connection has been set up this connection, it can be easily re-used.
 6. Once the connection has been established, open up the TEST treenode, then the Table node, and then navigate to the CUSTOMER table.
 7. Click on the CUSTOMER table and drag it over to the form designer
 8. Save the application to any desired directory
 9. Press the Run button on the toolbar, or F9 on the keyboard to run the application.
 10. Data is now displayed in the grid. Note that the data is displayed in a full-featured, AJAX-based datagrid.

CUST_NO	CUSTOMER	CONTACT_FIRST	CONTACT_LAST
1001	Signature Design	Dale J.	Wackerburg
1002	Dallas Technologies	Glen	Brown
1003	Buttle, Griffith and	James	Buttle
1004	Central Bank	Elizabeth	Brocket
1005	DT Systems, LTD.	Tai	Wu
1006	DataServe	Tomas	Bright
1007	Mrs. Beauvais		Mrs. Beauvais
1008	Anini Vacation	Leilani	Briggs
1009	Max	Max	
1010	MPM Corporation	Miwako	Miyamoto

15 rows

First Prev 1 2 3 4 5 6 7 8 9 10 Next Last

Figure 2 -- Data being displayed in a grid without any code being written.

Thus, with writing zero lines of code and by merely using existing VCL for PHP components, a developer can quickly and easily have access to data and get to work on the functionality of the application rather than worrying about the plumbing of connecting to and displaying data.

Enhancements to VCL for PHP

Templated Forms

Templated Forms are a major new feature for Delphi for PHP 2.0. Templated Forms enable PHP developers to develop applications using “tagging” similar to ASP.NET or Cold Fusion. For instance, developers can now create a templated form and add tags like the following:

```
<button:php id="Button1" style="z-index: 1000; width: 75px; height: 25px"></button:php>
```

into their HTML pages. Those HTML pages can be edited in the Visual Designer or in the text editor. This mode of development enables the decoupling of the PHP code from the HTML code so that developers can pass off their HTML files to external designers who can provide the styling of the HTML pages while working around the PHP tags in the HTML. This is a major feature for those PHP developers who are hesitant to begin using a pure visual design solution.

Zend Framework Support

Delphi for PHP 2.0 now includes the Zend Framework for PHP. The VCL for PHP includes four components that can be used to add the Zend Framework to an application.

Enhancement to the Delphi for PHP 2.0 IDE and Code Editor

Form Designer

The Form Designer has been completely rewritten for Version 2.0 with an emphasis on performance and smoothness. Developers can expect the designer to respond instantly to the placing and movement of components. Developers will find that placing and manipulating components on the designer is responsive and will enable them to layout their components with speed and precision. By using the Form Designer and the Object Inspector, developers can control all aspects of their PHP pages, including DocType, Encoding settings, colors, etc. The Form Designer also includes visual guidelines that aid in the alignment and spacing of code. As controls are moved on the design surface, the visual guidelines appear, indicating that the sides of selected controls are aligned. Controls also “snap” into alignment, making it easy for developers to align and size controls. In addition, as controls are moved next to each other, small guidelines appear between controls, enabling developers to properly space controls.

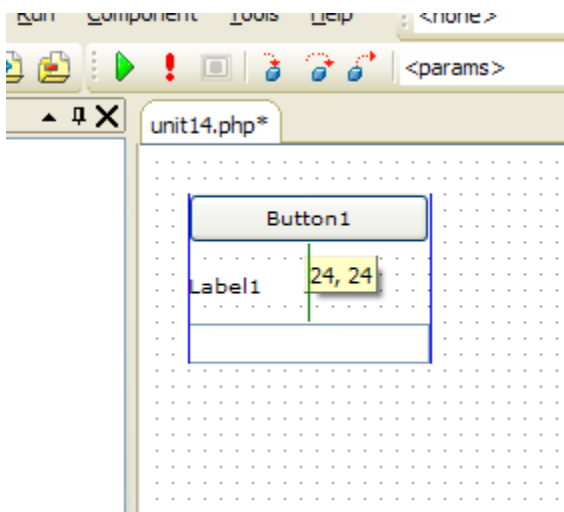


Figure 3 -- The Form Designer displaying alignment and spacing guidelines.

HTML Designer

In direct response to customer requests, Delphi for PHP 2.0's HTML Designer now enables the graphical development of HTML documents with embedded PHP and templated forms. HTML pages can be designed in a WYSIWYG fashion or in pure code. PHP can be embedded in the HTML files and made part of a project. Code Insight also works in the HTML designer, providing support for PHP while writing HTML and PHP code together. This is a powerful new feature that enables developers to have complete control over their HTML while at the same time having the power of RAD development.

Filterable Tool Palette

Finding components on the Tool Palette can sometimes be difficult, especially as the number of components increases. The Tool Palette now makes it very easy to find a specific component by filtering the components on the Tool Palette based on developer input. Developers can click the funnel icon to put the Tool Palette into filter mode, and then begin typing any portion of the name of the component they are looking for, and the Tool Palette will then display only those components whose name contains the text entered by the developer.

Code Folding

The editor now supports Code Folding, allowing developers to hide or reveal portions of their PHP code. Developers can choose to collapse function bodies, individual `<php>` tags, and other code segments so that they can concentrate on the code that they want to see.

SyncEdit

SyncEdit is a feature that enables developers to edit all the identical code symbols in a selected portion of code. The developer can select a portion of code, invoke SyncEdit, and the editor will then highlight all the identical identifiers in the portion of code under the SyncEdit mode. The developer can then tab from identifier to identifier, and make changes in one location which will result in the identifier being changed in all locations in the selected code. For instance, if the developer selects a chunk of code with the identifier "Button1" in three places, the developer can change "Button1" to "MyButton" in one location, but all the instances of "Button1" within the SyncEdit block will be changed at the same time. (Note that SyncEdit is purely syntactical and doesn't do anything more than symbol matching.)

Change Tracking

The Delphi for PHP 2.0 IDE will now show an indicator in the left side of the editor that tells the developer the state of change of a given line of code. A yellow bar indicates the line has changed since the last save. A green bar indicates that the line of code has changed since the file was first opened. Thus, developers can determine what portions of a given file have been altered.

Error Insight

Error Insight is a Code Editor feature that constantly scans source code, immediately marking syntax errors as they are found. Errors in the Code Editor are underlined with a red “squiggly” line, and are also displayed in the TreeView as part of the Structure Pane. (See Figure 4 below).

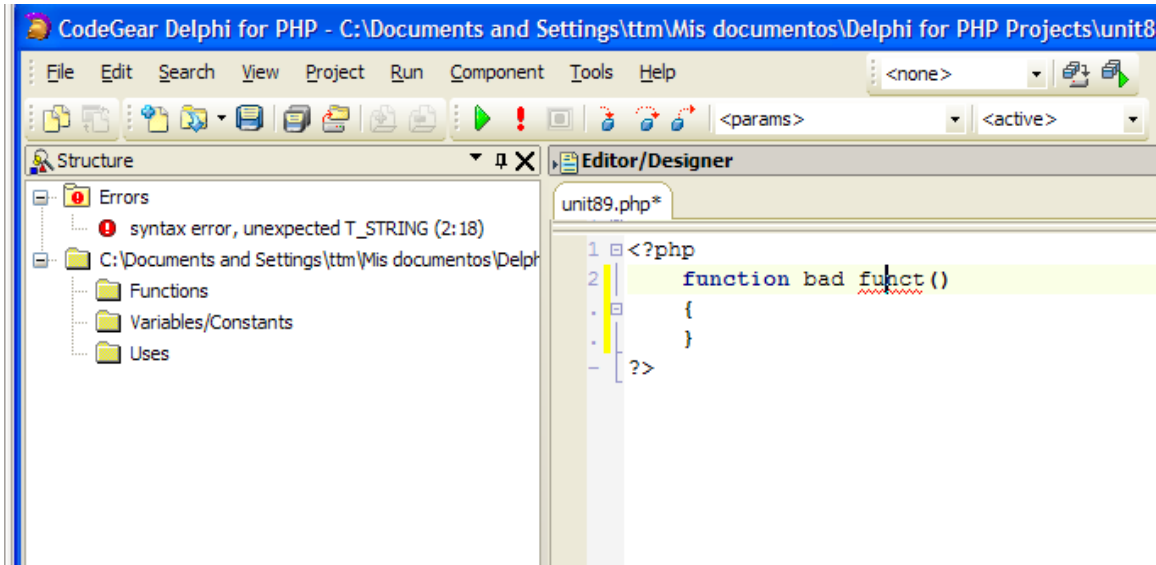


Figure 4 -- Error Insight showing and error both directly in the Code Editor as well as in the Structure Pane window.

Code Insight

Code Insight is a collection of Code Editor features that enable developers to type code more efficiently and effectively with fewer errors and with insight into the structure and capabilities of their code. It includes a number of separate features described below:

Code Completion

Code Completion is an IDE feature that provides developers with both a coding productivity tool and a view into the members of any given PHP class. As a developer types object code, a window pops up providing a list of the possible methods and members of the given object. (The Code Completion) Once the list appears (See 3), developers can either select the desired member, press Enter, and the selected member is automatically entered in the Code Editor. Alternately, the developer can continue typing the name of the desired identifier, and the list will be filtered to include only those members matching the entered text. At any time, the developer can press the enter key to have Code Insight enter the selected item.

Help Insight

In addition, Help Insight provides a Window attached to the Code Completion window that provides basic help information about the identifier selected in the Code Completion window. As shown in Figure 3, Help Insight for the GetOnClick method is displayed,

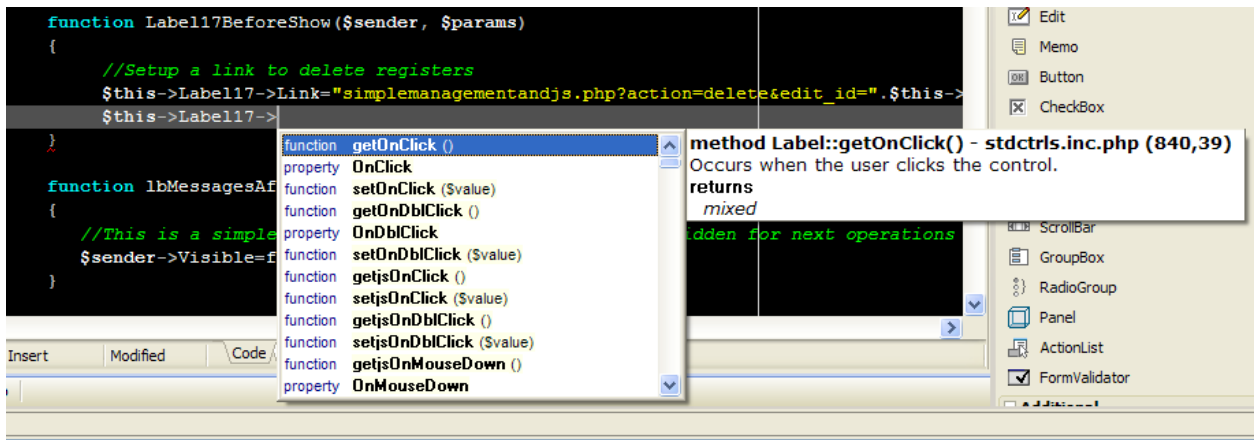


Figure 5 -- The IDE displaying CodeInsight and HelpInsight for a Label control

including a short description of its function, what value it returns, and where it can be found in the VCL for PHP source code. These Help Insight hints result from the inclusion of [PHPDoc](#) comments in the source code.

Parameter Insight

When coding, developers naturally make calls to routines requiring parameters, and remembering the exact order and type of those parameters isn't always possible. Delphi

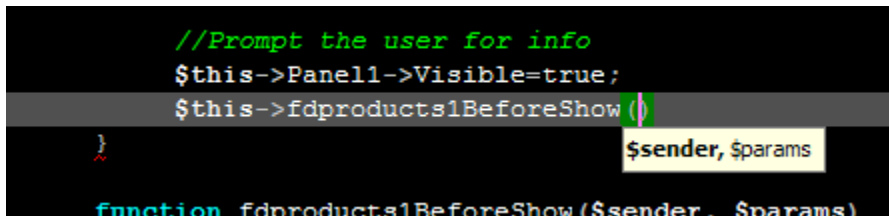


Figure 6 -- The Code Editor providing Parameter Insight, a hint window that informs the developers about the parameters required by the current routine.

for PHP 2.0 provides Parameter Insight, a feature that provides a small hint window that informs the developer about the currently required parameters. As parameters are entered, the hint window changes by making bold the current parameter that contains the text cursor. Delphi for PHP 2.0 also uses the available PHPDoc to provide typing information in Parameter Insight. This is a useful and powerful feature. The combination of PHPDoc information and Parameter Insight enables developers to write more efficient code.

Other Delphi for PHP 2.0 Enhancements

Wiki-based Documentation

All the documentation for Delphi for PHP 2.0 is now hosted on a Wiki, allowing community members to add and update content. In addition to the content being constantly updated, developers can download the help from the Wiki at any time for use on a local computer.

Internationalized IDE

Delphi for PHP 2.0's IDE is completely localizable by design. Delphi for PHP 2.0 ships with the ability to display the IDE in the following languages: English, Spanish, German, Traditional Chinese, and Portuguese. In addition, the string tables used by the IDE are laid out in such a way that they can be easily translated, and those translations can be easily shared by community members for the benefit of all. In addition, Delphi for PHP 2.0 includes a tool to translate IDE strings to new languages. In addition, the PHP Help manual, the Welcome page and the IDE help are in sync with the active language and are external resources that can be translated without requiring an IDE update.

Improved Database Access

Delphi for PHP 2.0 includes new and improved support for component-based access to Oracle, Microsoft SQL Server, PostgreSQL, DB2, Informix, and Sybase.

Delphi for PHP 2.0 in Depth

The Delphi for PHP 2.0 Integrated Development Environment

General Notes

The IDE in Delphi for PHP 2.0 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, Delphi for PHP 2.0's Code Editor provides extensive support for typing and numerous aids for code creation that greatly enhance a developer's productivity. PHP developers require a powerful, productive Code Editor, and Delphi for PHP 2.0 provides it. Among its many features are:

- 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.
- Code Insight produces pop-up windows that give insight into parameters needed for a given routine.
- Error Insight provides immediate feedback by underlining code syntax errors.
- Help Insight provides pop-up windows right in the editor giving basic documentation and declaration information about any identifier within code.

- Parenthesis and Brace Matching provide visual clues as to what opening and closing identifiers match up with each other
- Fast and easy code navigation via a CTRL-Click mechanism. The declaration of identifiers can be easily located by selecting CTRL-Click on any identifier. This is a particularly useful feature given the disparate nature of PHP code.
- The Source Code Formatter formats PHP code to a user-specified format on demand.
- Code Templates allow the developer to create common coding constructs and quickly add them to code with a CTRL+J keystroke

The Visual Designer

When not writing code, much of a developer's time will be spent laying out forms with components to create a user interface. The IDE provides a powerful Form Designer to do just that. Delphi for PHP 2.0's Form Designer looks exactly like a web page, enabling What-You-See-Is-What-You-Get (WYSIWYG) layout of forms at design-time.

Developers can drag-and-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.

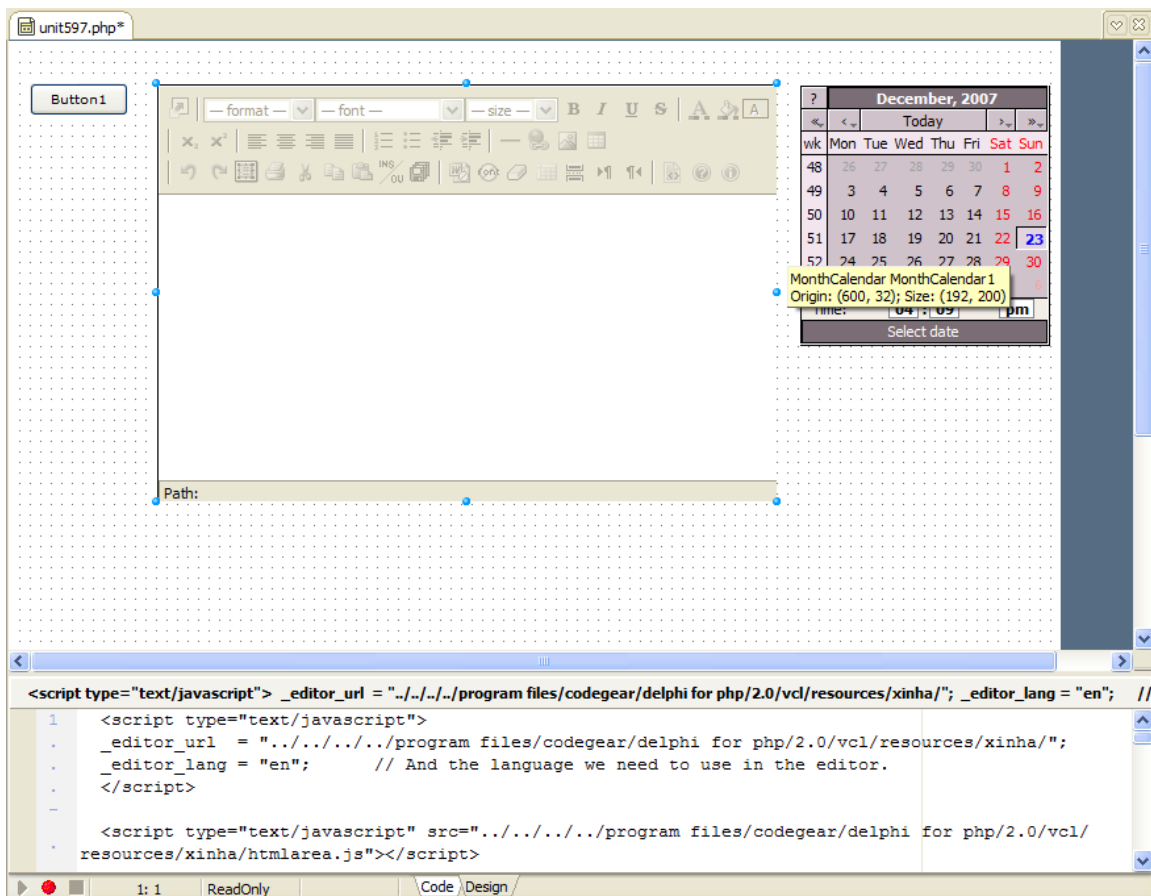


Figure 7 -- The Visual Designer with the Tag Editor enables a high level of control over the layout of web pages.

Once components are placed on the form, their properties can be set using the Object Inspector. Listing all a component's properties and events, the Object Inspector allows a developer to quickly and easily manipulate a component's 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, OnShow, etc. – occur to the component. In addition to server-side, PHP-driven events, developers can use the designer to add JavaScript events to any page element.

The Tool Palette houses all the VCL components installed into the IDE. It includes by default all the standard components that ship with Delphi for PHP 2.0. In addition, developers can add in components from third-parties.

The Project Manager

Applications quickly become complicated with numerous forms and code files. Many applications consist of any number of different binaries and projects. Delphi for PHP 2.0'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. Existing files and forms can be opened in the IDE. 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.

The Data Explorer

The Data Explorer provides easy access to database connectivity. Developers can add connections to any of their databases in the Data Explorer. They can then create instances of those connections on their web pages via a simple drag-n-drop process. When a table is dragged onto the design surface, the designer automatically creates a control to display the data, using a DBGrid, a DBPaginator, or a DBRepeater. (The choices are user configurable). Likewise, when a table's field is dragged to the designer, the developer can choose what type of control is used to display the field on the web page. This capability allows for lightning fast connectivity to data on a web page. Developers can literally be manipulating data display in just a few mouse actions.

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. Delphi for PHP 2.0 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. 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. 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.

Additional Debugger Features

The Profiler

Delphi for PHP 2.0's debugger includes a profiler. As an application is run under the control of the debugger, each line of code is timed as it executes, and the profiler records this information for later analysis. As shown in Figure 8, the profiler tracks the number of times a line of code is executed and keeps a running total and calculates an average execution time. With this information, developers can easily spot bottlenecks and inefficiencies in their code and target them for optimization.

File	Hits	Avg Time (ms)	Total (ms)	Min (ms)	Max (ms)	Path
TestCanvas.php(2)	1	5.842	5.842	5.842	5.842	C:\Program Files\Code
TestCanvas.php(3)	1	0.021	0.021	0.021	0.021	C:\Program Files\Code
TestCanvas.php(4)	1	0.026	0.026	0.026	0.026	C:\Program Files\Code
TestCanvas.php(5)	1	0.014	0.014	0.014	0.014	C:\Program Files\Code
TestCanvas.php(10)	1	0.038	0.038	0.038	0.038	C:\Program Files\Code
TestCanvas.php(14)	1	0.007	0.007	0.007	0.007	C:\Program Files\Code
vd.inc.php(29)	1	0.034	0.034	0.034	0.034	C:\Program Files\Code
vd.inc.php(33)	1	0.014	0.014	0.014	0.014	C:\Program Files\Code
vd.inc.php(38)	1	0.034	0.034	0.034	0.034	C:\Program Files\Code
vd.inc.php(40)	1	7.828	7.828	7.828	7.828	C:\Program Files\Code
vd.inc.php(42)	1	0.015	0.015	0.015	0.015	C:\Program Files\Code
vd.inc.php(44)	1	0.026	0.026	0.026	0.026	C:\Program Files\Code
vd.inc.php(53)	1	2.132	2.132	2.132	2.132	C:\Program Files\Code
vd.inc.php(54)	1	0.015	0.015	0.015	0.015	C:\Program Files\Code
vd.inc.php(57)	1	0.03	0.03	0.03	0.03	C:\Program Files\Code
vd.inc.php(59)	1	0.03	0.03	0.03	0.03	C:\Program Files\Code
vd.inc.php(66)	1	0.026	0.026	0.026	0.026	C:\Program Files\Code
vd.inc.php(72)	1	0.016	0.016	0.016	0.016	C:\Program Files\Code
vd.inc.php(85)	1	0.007	0.007	0.007	0.007	C:\Program Files\Code
vd.inc.php(87)	1	0.055	0.055	0.055	0.055	C:\Program Files\Code

Figure 8 -- The profiler measuring the performance of each line of code in a PHP project

Local Variables

During the execution of an application, the debugger makes frequent stops and breakpoints as the application is debugged. One of the reasons to set a breakpoint is to be able to examine the state of the application when the breakpoint is arrived at. The Local Variables view allows the developer to do just that. As shown in Figure 9, it provides a tree-structured view of the existing local variables that are in scope for the given breakpoint. Developers can examine the current state of both the PHP and the existing request for the page being executed.

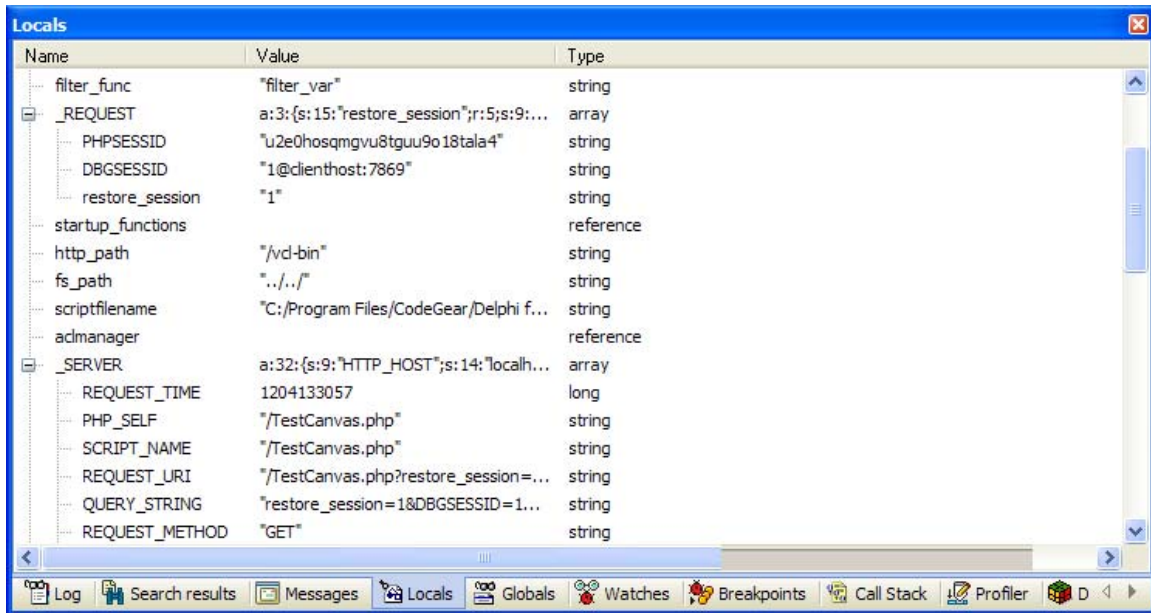


Figure 9 -- The Local Variables view under the debugger

The Visual Component Library (VCL) for PHP 2.0

VCL for PHP is a 100% PHP-based component framework that enables developers to design web pages visually. Inspired by the VCL in Delphi, it is deeply integrated into the Delphi for PHP 2.0 IDE. It provides a set of visual components that allow RAD development of web pages, web sites, and web applications. The VCL for PHP is expandable and can be used to build any component that can be rendered using HTML, AJAX, JavaScript or any other web-based presentation technology.

The VCL for PHP contains, as its base, components that encapsulate all the basic HTML form components such as buttons, edit boxes, list boxes, checkboxes, etc. These types of controls make up the basic interface of most web applications. The VCL for PHP, in conjunction with the form designer, knows how to emit the proper HTML for these controls as properties are set in the Object Inspector for the given control.

The VCL for PHP is an extensible framework. VCL for PHP components can be written by inheriting from existing classes in the framework. These base classes, such as TComponent, are all written in PHP and provide the infrastructure for generating HTML, JavaScript, and PHP code as desired by the component writer. Developers can build components that output anything that can be expressed in a browser, providing great power by encapsulating any functionality in a reusable, visual control.

In addition, the VCL for PHP leverages this extensibility by encapsulating the most popular libraries and scripts from the PHP community and marketplace, including:

- Zend (<http://framework.zend.com>)
- qooxdoo (<http://qooxdoo.org/>)

- ADOdb (<http://adodb.sourceforge.net>)
- DynAPI (<http://dynapi.sourceforge.net/>)
- Smarty (<http://smarty.php.net/>)
- xajax (<http://www.xajaxproject.org/>)
- JSCalendar (<http://www.dynarch.com/projects/calendar>)

The components model in VCL for PHP contains:

- Properties – properties allow developers to make changes to specific attributes of a component that may have code attached to the changing of the property. For instance, changing the Caption property of a button may change the Width property as well.
- Methods – Methods are functions or procedures that are encapsulated in the functionality of the component
- Events – Events are specialized methods that are executed according to the behavior or specific occurrences that happen to the object. Events contain code that is run on the server-side.
- JavaScript Events – JavaScript events are similar to server-side events, except that they execute on the client-side and written as JavaScript

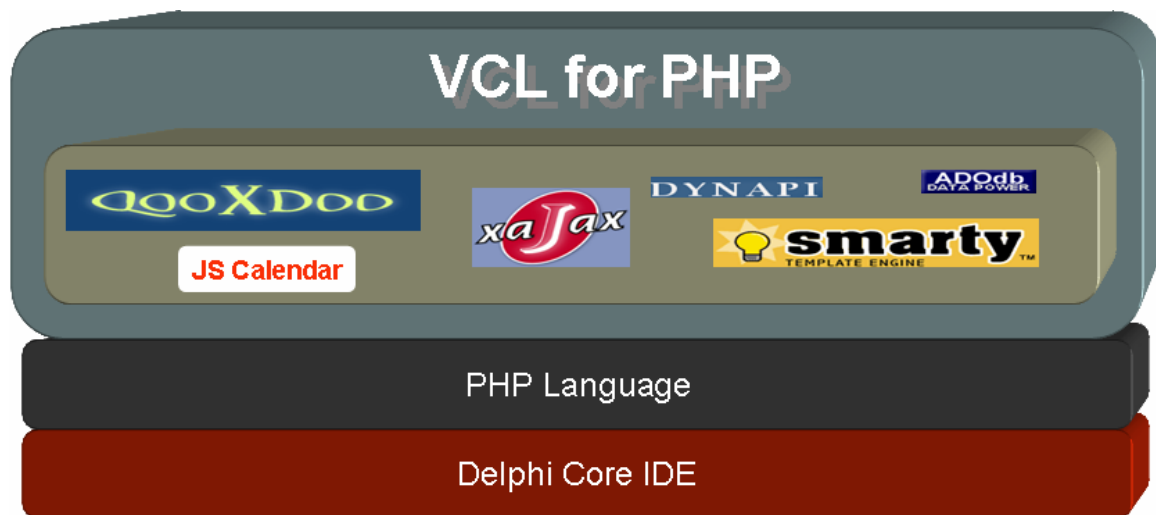


Figure 10 -- The VCL for PHP structure, including integrated third-party frameworks and libraries.

VCL for PHP is Open Source and available at SourceForge:

<http://vcl4php.sourceforge.net>

This means the PHP developer community is able to update existing code, fix bugs, and create new components for VCL for PHP.

Database and VCL for PHP

Database development can be performed through the use of Data Access components. They are integrated to the ADOdb for PHP library, which are a set of components for database development and which allow access to many different databases, including:

- InterBase
- MySQL
- Access
- ADO, ODBC and OLEDB provider
- FoxPro
- FrontBase
- DB2
- Informix 7.x
- SQL Server 7, 2000 and 2005
- Firebird
- Oracle 8, 9
- PostgreSQL 6.x, 7 and 8
- SAP DB
- SQLite
- SQL Anywhere
- Sybase

Because data access is component based, Delphi for PHP 2.0 developers can access and display data in a web page without writing any code. Because the VCL for PHP components completely encapsulate the process of accessing data, developers can very easily get access to their data and focus on the application using that data instead of the code that accesses it.

Building a Simple Database Application

To create a simple database application that connects to an InterBase database, follow these simple steps.

1. Choose **File | New VCL for PHP Application** from the main menu. This creates a new form and opens it in the **Designer** on the **Design** tab.
2. Expand the **Data Access** category on the **Tool Palette** and select the **Database** component.
3. Click anywhere on the form to add the **Database** component to the application. A **Database** icon displays on the form grid and is selected as the active component.
4. Modify the following properties for the **Database** component in the **Object Inspector**:
 - a. Set the **DatabaseName** to an existing database using the form:
server:pathdatabase.gdb (On a default install, the value *localhost*:

*C:\Program Files\Common Files\CodeGear
Shared\Data\EMPLOYEE.GDB* should work fine).

- b. Set the **DriverName** property by selecting the database driver from the drop down list, for example, **Borland_ibase** or **mysql**.
 - c. Set the **User** property the user name on the database server (If using the default table above, the username is “SYSDBA”).
 - d. Set the **Password** property on the database component. (If using the default database, this value is “masterkey”)
5. Add a **Query** component to the form and set the following properties in the **Object Inspector**:
- a. Set the **Database** property to *Database1*; It can be selected from the drop-down list on the property value.
 - b. Set the **Query** property to a valid SQL statement (If using the default database above, this property can be set to *select * from employee.*)
 - c. Set the **Active** property to *true*.
6. Add a **DataSource** component to the form and set the following property
- a. Set the **DataSet** property to *Query1*. This can be chosen from the drop down list on the property value.
7. Expand the **Data Controls** category on the **Tool Palette** and select the **DBGrid** component.
8. Add the "DBGrid" component to the form and set the following property:
- a. **Set the DataSource** property to refer to the **DataSource1** control.
9. Finally, click the **Run** button on the main toolbar or press **F9**.

The query data from the selected database should be visible on the resulting web page.

Additional Materials

The Delphi for PHP 2.0 Documentation Wiki

The Delphi for PHP 2.0 Documentation Wiki contains a number of screencasts that demonstrate various features of Delphi for PHP 2.0 and the VCL for PHP.

- [Hello, World!](#): Shows how to create a simple Hello, World application
- [Welcome page](#): How to use the welcome page and all its options
- [Tool Palette](#): How to insert components on forms, how to find components and how to use the keyboard
- [Object Inspector](#): Properties, Events, JavaScript events, subproperties, references, property editors, etc
- [Project Manager](#): Creating projects, project groups, managing files, add folder contents to project
- [Code Explorer](#): Browse for items, functions, updating code, PHPDoc support
- [Data Explorer and Data Viewer](#): Adding databases, modifying connections, different types of databases, browsing tables, executing queries
- [Docking and Desktops](#): Docking interface, desktop saving, debug desktop

- [Multilanguage](#): Change active language, request PHP help, localized welcome page
- [Forms](#): Create a simple application interface, two way tools
- [Templated Forms](#): Create an application interface using some HTML template
- [Component development](#): Create a basic component, including packages, icons for components, property editors, etc
- [Code Editor](#) Code Insight, code editor properties, syntax highlighting, code folding, sync ranges, sync editing. bookmarks, tracking band, macros, etc
- [Source Code Formatter](#): Options, before and after
- [Debugging](#): Watches, profiler, breakpoints, etc
- [Deployment](#): Deploying a project