



Tech Notes

Everything PHP Developers Need to Be Productive

Introducing Delphi for PHP 2.0

Andreano Lanusse, Embarcadero Technologies

February 2009

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

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

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

EVERYTHING PHP DEVELOPERS NEED TO BE PRODUCTIVE

Most PHP tools of today are pretty minimal and seem to go on the assumption that PHP developers like spending time constantly looking for “add-on” tools to improve their productivity -- such as HTML editors, plug-ins for their source code editors, better debuggers, and performance monitors. In addition, most tools on the market today don’t provide much on the way of managing frameworks and libraries. Many PHP developers change frameworks frequently and the number of frameworks and libraries for PHP increases every day. It takes a more sophisticated tool to manage these types of changes. Recognizing that PHP developers need the same kinds of robust IDEs available for other languages, Embarcadero introduces Delphi for PHP 2.0 to address the requirements of PHP community.

This article discusses Delphi for PHP 2.0’s indispensable features that help PHP s write PHP applications faster and with higher quality. At the end of this article you won’t need to start a new project to use Delphi for PHP, you will just open your PHP files and will be able to start using the source code editor, debugger, profiler, and visual development.

GENERAL OVERVIEW OF DELPHI FOR PHP 2.0

Delphi for PHP 2.0 is a rapid application development (RAD) - based integrated development environment (IDE) for developing pure 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-and-drop development environment. PHP developers can now enjoy the same productivity gains 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 a powerful and productive editing environment that PHP developers who like to “hand-tweak” their PHP code have come to expect. Delphi for PHP 2.0 aims to increase developers’ productivity with both a RAD environment and a robust code editor.

BUILDING A VISUAL DEVELOPMENT

FORM DESIGNER

When not writing code, a developer spends much of his time creating user interfaces. Delphi’s 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. You 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.

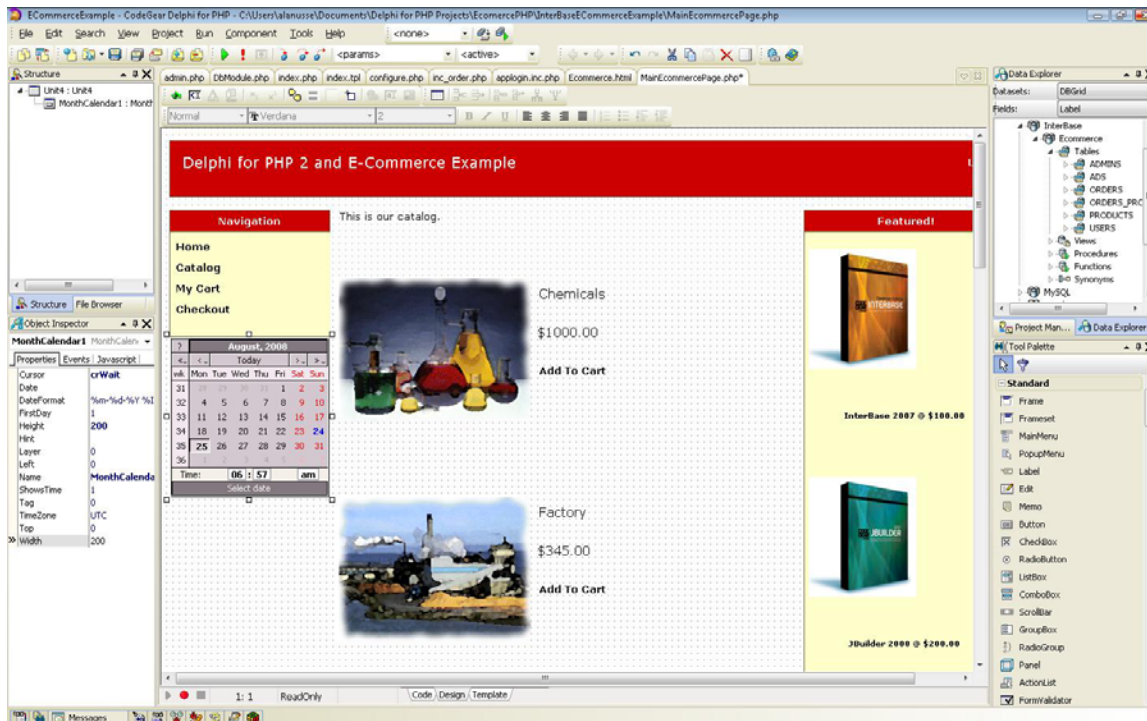


Figure 1 the Form Designer with the Tag Editor enables a high level of control over the layout of web pages.

Once components are placed on the form, their properties can be set using the Object Inspector. Listing all a components properties and events, the Object Inspector allows you to quickly and easily manipulate a component's appearance, and to attach code to the various events that might occur on that component. The Object Inspector also allows you to 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 run event-driven code on events such as OnClick, OnShow, and so on. In addition to using server-side, PHP-driven events, you can use the designer to add JavaScript events to any page element.

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 you to align and size controls. In addition, as controls are moved next to each other, small guidelines appear between controls, letting you fine tune the spacing between controls.

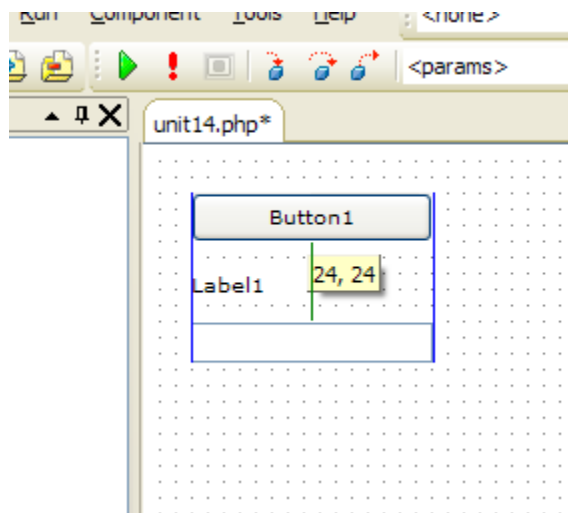


Figure 2 the Form Designer displaying alignment and spacing guidelines.

HTML DESIGNER

The HTML Designer enables graphical development of HTML documents with embedded PHP and templated forms. HTML pages can be designed in a WYSIWYG view or in pure code. PHP can be embedded in the HTML files and incorporated into a project. Code Insight also works in the HTML designer, providing support for PHP while writing both HTML and PHP code. This is a powerful new feature gives you complete control over your HTML while at the same time having the power of RAD development.

TEMPLATED FORMS

Templated Forms enables “tagging” similar to ASP.NET or Cold Fusion. For example, you can 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 HTML pages. Those HTML pages can be edited in the Visual Designer or with 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 advantage for PHP developers who might be hesitant to use a visual design solution.

THE VISUAL COMPONENT LIBRARY (VCL) FOR PHP

The VCL for PHP 2.0 leverages this extensibility by encapsulating the most popular libraries and scripts from the PHP community and marketplace, including:

- Zend (<http://framework.zend.com>)
- Oracle native component
- MySQL native component

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.

ZEND FRAMEWORK SUPPORT

Delphi for PHP 2.0 includes the Zend Framework for PHP. The VCL for PHP includes four components that add the Zend Framework to an application. Those components allow you to visually manage the authentication process for any PHP application through user and roles. For each user or group, you can define the rights to access each components or class in your PHP page.

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 is a set of components for database development and supports access to many different databases, including:

- InterBase
- MySQL
- Access
- ADO, ODBC and OLEDB provider
- FoxPro
- FrontBase
- DB2
- Informix
- SQL Server
- Firebird
- Oracle
- PostgreSQL
- SAP DB
- SQLite
- SQL Anywhere
- Sybase

Because data access is component-based, you can access MySQL and Oracle through the VCL components using native access, improving the performance for both databases.

POWERFUL PHP SOURCE CODE EDITOR

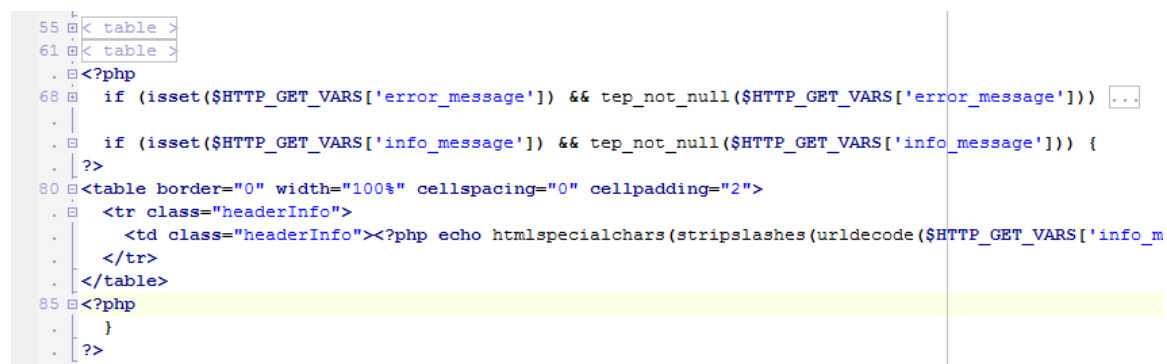
No mere text editor, Delphi for PHP 2.0's Code Editor provides extensive support for typing and numerous aids for code creation that greatly enhance your productivity. PHP developers require a powerful, productive code editor, and Delphi for PHP 2.0 provides it. Among its many features are:

- 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 you type.
- 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 feature is particularly useful given the disparate nature of PHP code.
- The Source Code Formatter formats PHP code to a user-specified format on demand.
- Code Templates allow you to create common coding constructs and quickly add them to code with a Ctrl +J keystroke.

CODE FOLDING

You can hide or reveal portions of your PHP code with Code Folding. Code Folding allows you to collapse function bodies, individual `<php>` tags, and other code segments so that they can concentrate on only the code that they want to see.



```
55 <table >
61 <table >
. <?php
.   if (isset($_HTTP_GET_VARS['error_message']) && tep_not_null($_HTTP_GET_VARS['error_message'])) ...
.   if (isset($_HTTP_GET_VARS['info_message']) && tep_not_null($_HTTP_GET_VARS['info_message'])) {
.   ?>
80 <table border="0" width="100%" cellspacing="0" cellpadding="2">
.   <tr class="headerInfo">
.       <td class="headerInfo"><?php echo htmlspecialchars(stripslashes(urldecode($_HTTP_GET_VARS['info_m
.   </tr>
.   </table>
85 <?php
.   }
.   ?>
```

Figure 3 Code Folding in PHP file.

SYNCEdit

SyncEdit lets you edit all identical code symbols in a selected portion of code. The developer selects a portion of code, invokes SyncEdit, and the editor then highlights all the identical identifiers in the selected portion of code. You can then tab from identifier to identifier, and make changes in one location that result in the identifier being changed in all locations in the selected code. For example, if you select a chunk of code with the identifier "Button1" in three places, you 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.)

```

. // check if the 'install' directory exists, and warn of its existence
. if (WARN_INSTALL_EXISTENCE == 'true') {
.     if (file_exists(dirname($HTTP_SERVER_VARS['SCRIPT_FILENAME']) . '/install')) {
16 |     $messageStackNEW->add('header', WARNING_INSTALL_DIRECTORY_EXISTS, 'warning');
.     }
.     }
.
. // check if the configure.php file is writeable
20 | if (WARN_CONFIG_WRITEABLE == 'true') {
.     if ( (file_exists(dirname($HTTP_SERVER_VARS['SCRIPT_FILENAME']) . '/includes/co
.     $messageStackNEW->add('header', WARNING_CONFIG_FILE_WRITEABLE, 'warning');
.     }
.     }
.
. // check if the session folder is writeable
. if (WARN_SESSION_DIRECTORY_NOT_WRITEABLE == 'true') {
.     if (STORE_SESSIONS == '') {
30 |     if (!is_dir(tep_session_save_path())) {
.         $messageStackNEW->add('header', WARNING_SESSION_DIRECTORY_NON_EXISTENT, 'wa
.     } elseif (!is_writeable(tep_session_save_path())) {
.         $messageStackNEW->add('header', WARNING_SESSION_DIRECTORY_NOT_WRITEABLE, '
.     }
.     }
. }

```

Figure 4 Sync Edit in action.

CHANGE TRACKING

The Delphi for PHP 2.0 IDE shows an indicator on the left side of the editor that tells you the state 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, you can determine what portions of a given file have been altered.

```

. // check if the 'install' directory exists, and warn of its existence
. if (WARN_INSTALL_EXISTENCE == 'true') {
.     if (file_exists(dirname($HTTP_SERVER_VARS['SCRIPT_FILENAME']) . '/install'))
.         $messageStack->add('header', WARNING_INSTALL_DIRECTORY_EXISTS, 'warning');
17 |     //This line is modified
.     }
.     }
. }

```

Figure 5 Change Tracking

ERROR INSIGHT

Error Insight is a Code Editor feature that constantly scans source code, 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 6.

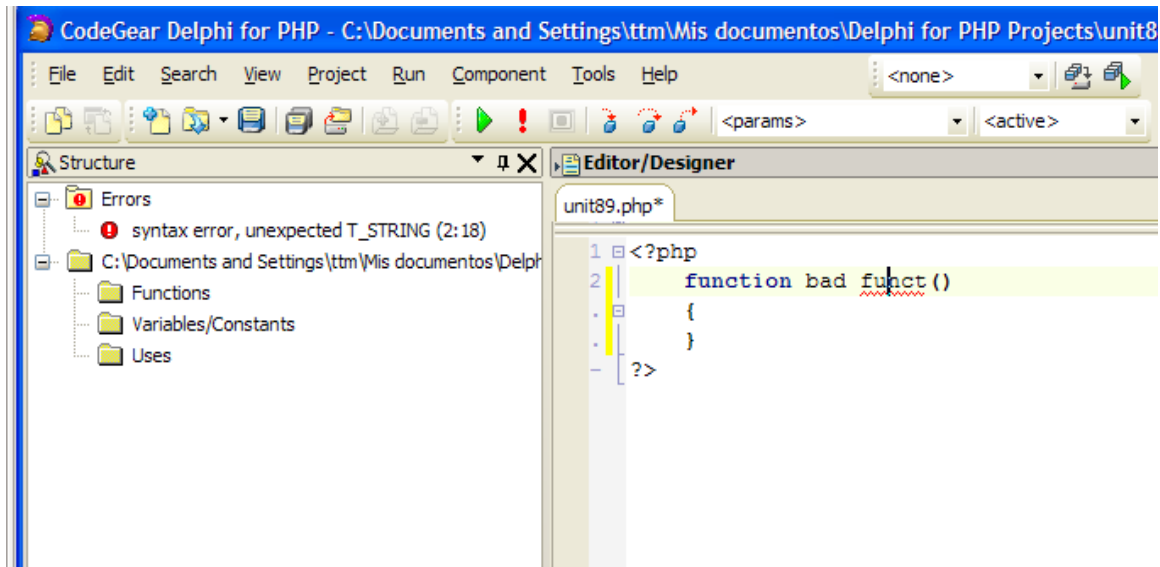


Figure 6 Error Insight showing and error both directly in the Code Editor as well as in the Structure Pane window.

ENCODING, FORMAT AND CHARACTER SET

You can specify the encoding, the format and character set for your files. To do this, you can right-click source code to specify options for the active source code or use Tools | Options | Editor Options to set up the default settings for all files.

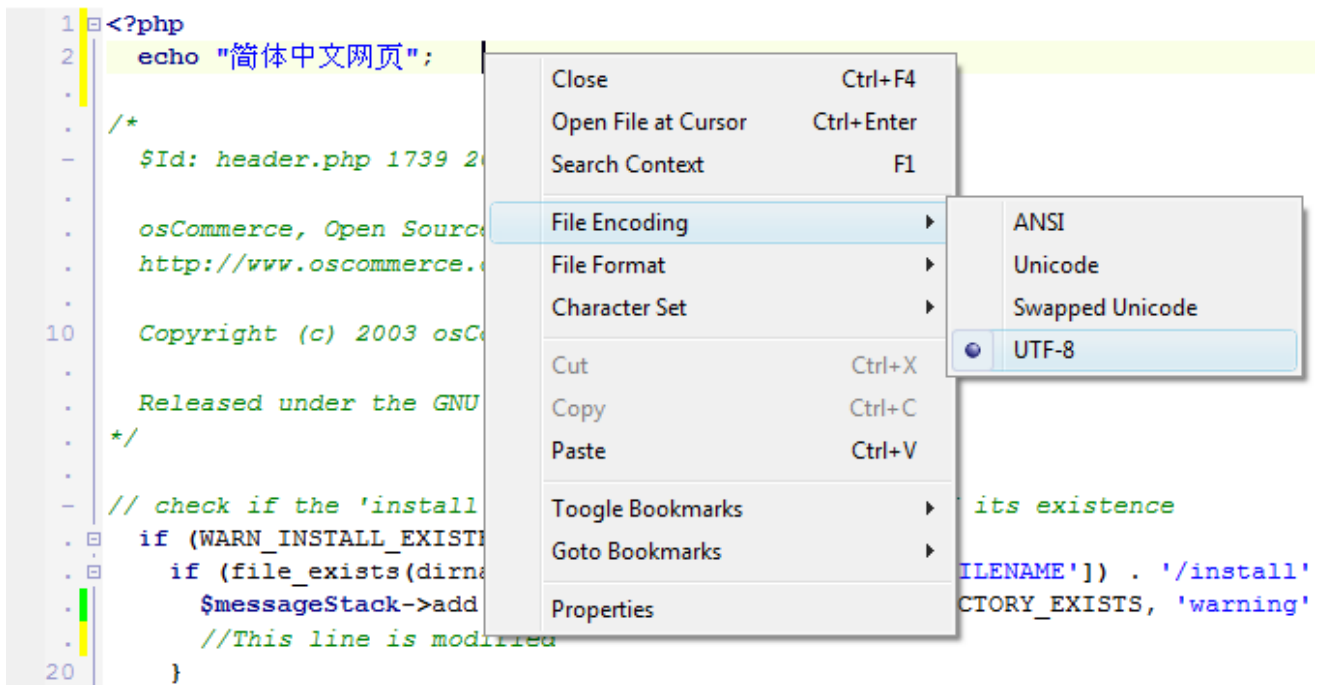


Figure 7 encoding a file as UTF-8.

CODE INSIGHT

Code Insight is a collection of Code Editor features that enable you 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.

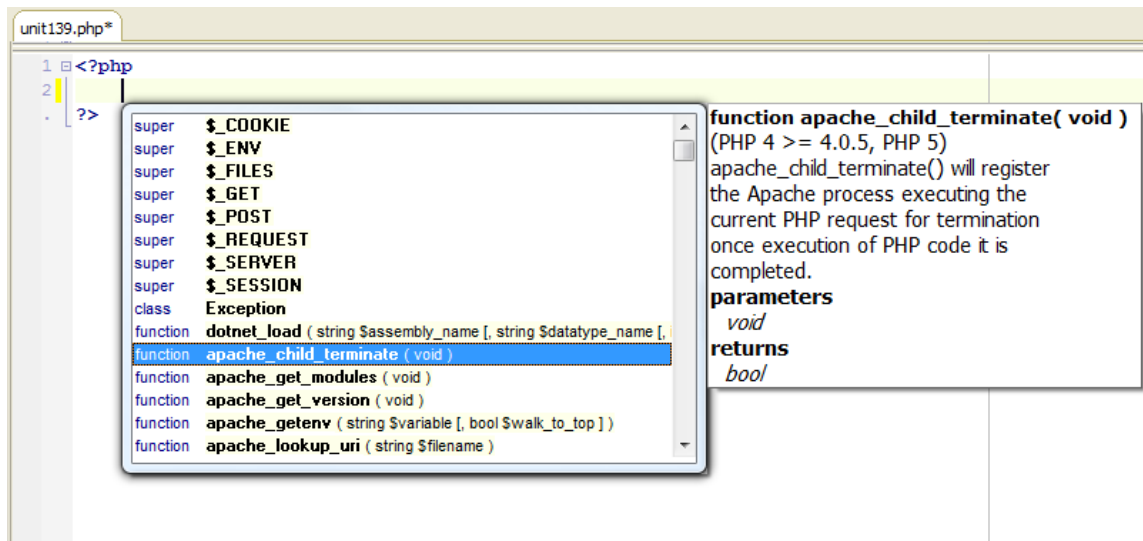


Figure 8 Code Insight and PHP documentation for the function `apache_child_terminate`.

CODE COMPLETION

Code Completion is both a coding productivity tool and a view into the members of any given PHP class. As you write object code, a window pops up providing a list of the possible methods and members of the given object. Once the list appears (see Figure 7), you select the desired member and press Enter, and the selected member is automatically entered in the Code Editor. Alternately, you 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, you 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 8, Help Insight for the `GetOnClick` method is displayed, 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.

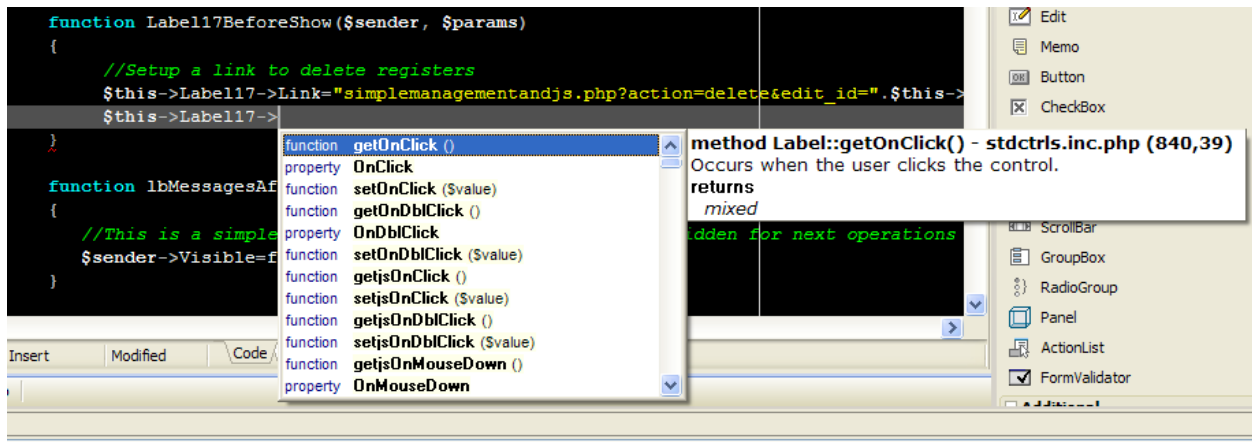


Figure 9 the IDE displaying CodeInsight and HelpInsight for a Label control.

PARAMETER INSIGHT

When coding, you naturally make calls to routines requiring parameters, and remembering the exact order and type of those parameters can be challenging.

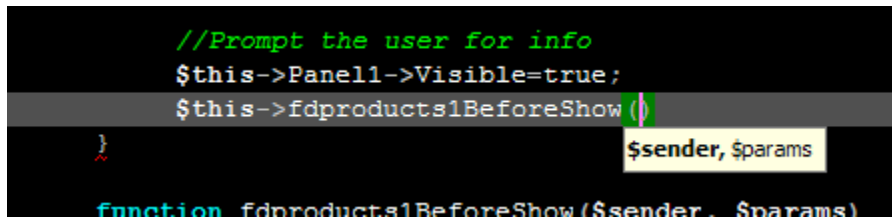


Figure 10 The Code Editor providing Parameter Insight, a hint window that informs developers about the parameters required by the current routine.

Delphi for PHP 2.0 provides Parameter Insight, a feature that provides a small hint window that informs you about the currently required parameters. As parameters are entered, the hint window changes by making the current parameter bold. 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 you to write more efficient code.

BOOKMARKS

If you need to return to a location in your source code, you can set a bookmark, so you don't need to search again for that specific location. You have up to 10 bookmarks. To set a bookmark in a specific location, use Shift+Ctrl+N, where N is a number from 0 to 9. To return to a bookmark, use Ctrl+N. You can also right-click the source code and use the popup menu to toggle a bookmark at a specific location or to go to a bookmark.

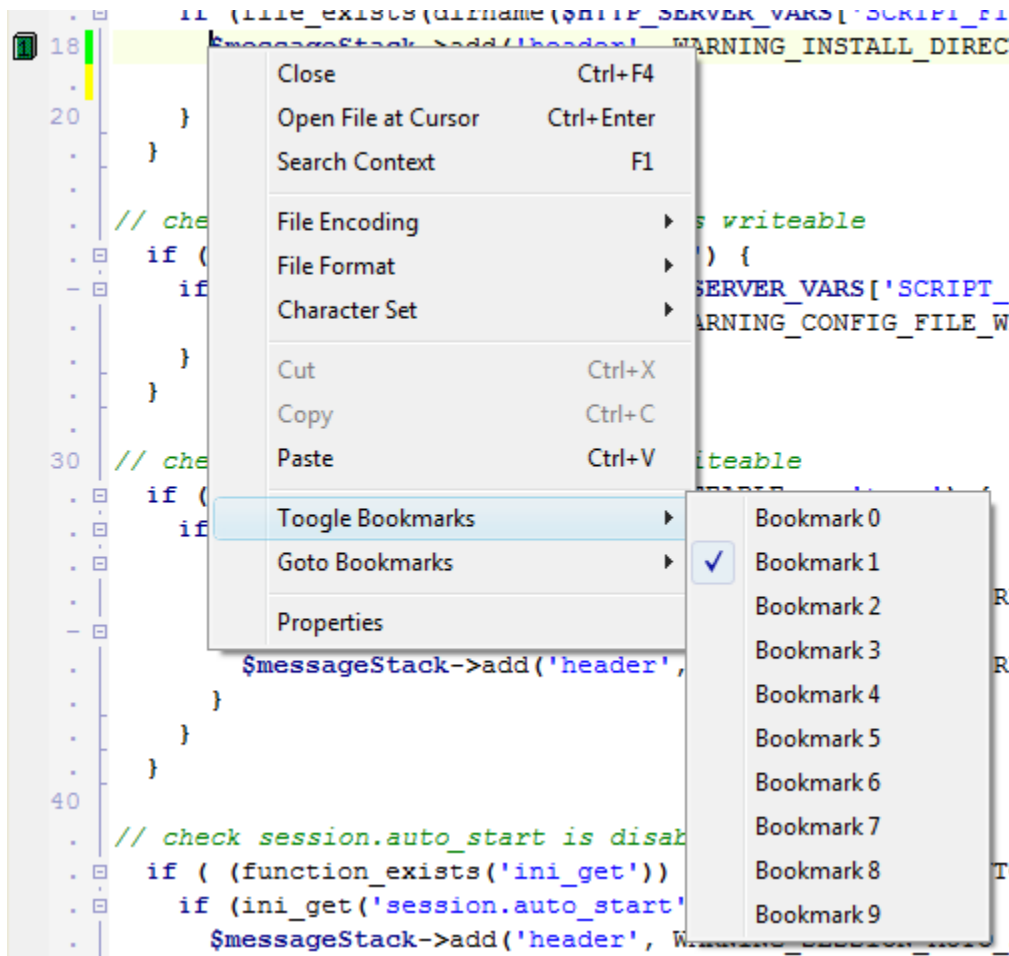


Figure 11 Toggling to Bookmark 1

CUSTOMIZABLE KEYMAPPINGS AND SOURCE CODE FORMATTER

You can customize most of the code editor actions to use your own keys, and create your custom key mapping; so the editor adapts to your way of work.

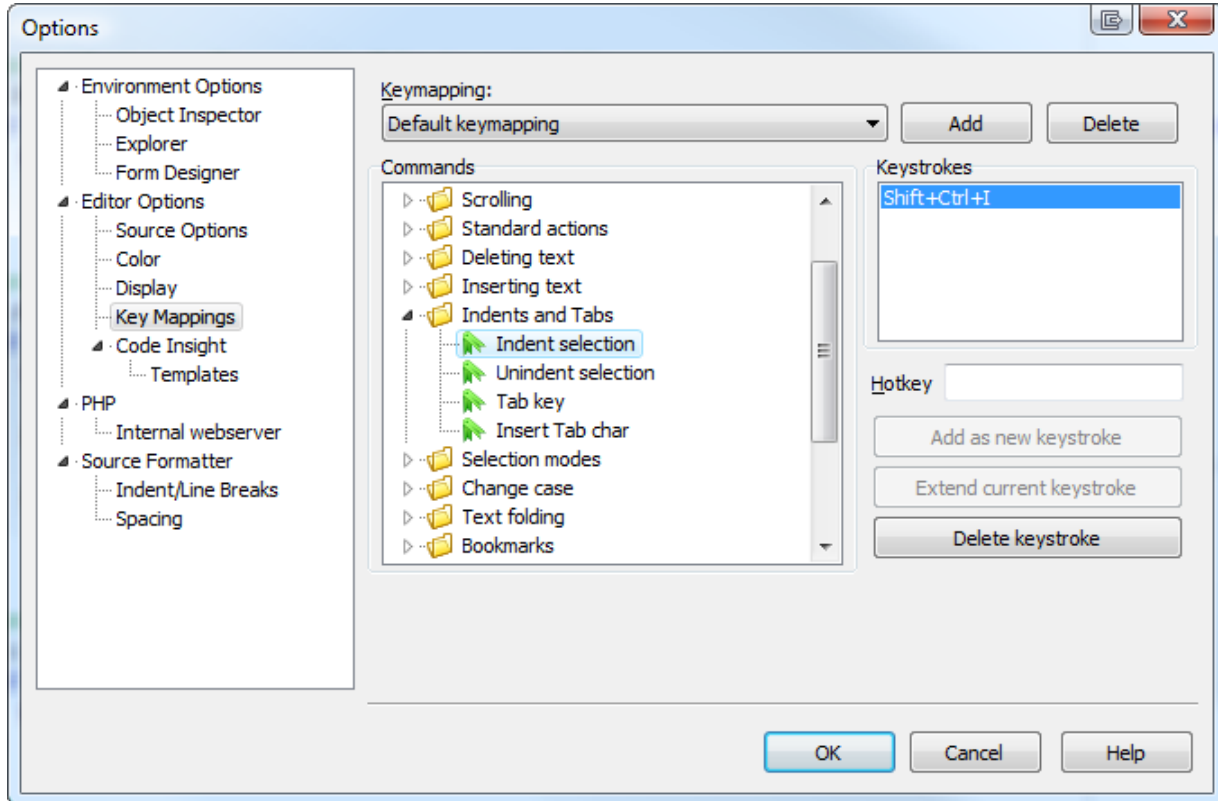


Figure 12 Keymapping windows, customizing the Indent selection.

If you need to modify source code from another developer, or simply want to keep your source code organized and don't want to mess with indentation, you can use the Source Formatter to re-organize your source code. With the Source Formatter, you can customize how your source code is formatted.

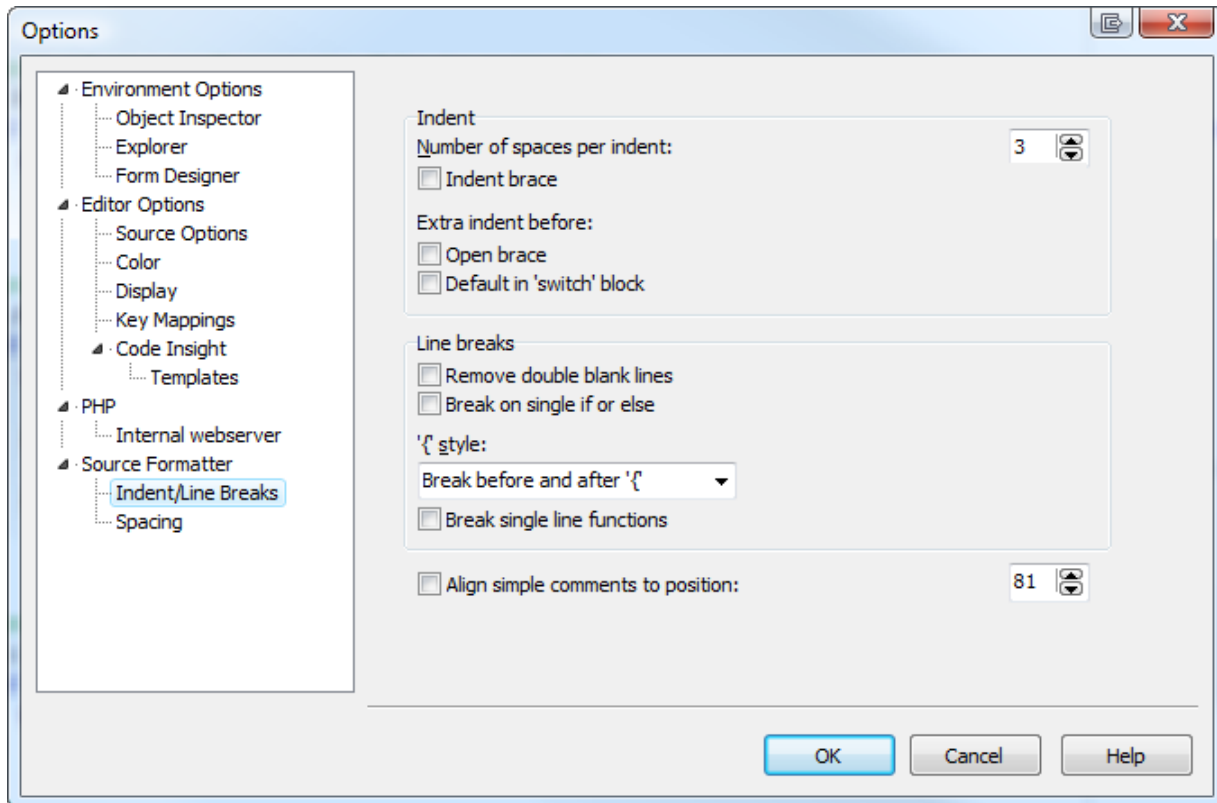


Figure 13 Source Code editor options.

STRUCTURE PANE

The Structure Pane works in different ways depending on the type of source code you are editing. For PHP code, it shows a structured view of the classes, functions, methods, fields, includes, and so on. If you double-click an element, the IDE opens the source code at the specific location the element was defined. If you edit HTML, JS or CSS files, the Structure Pane shows the structure of the document you are writing.

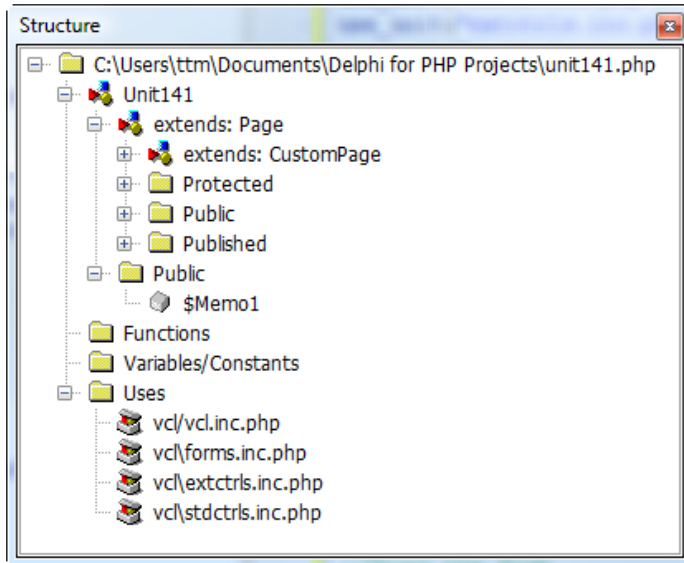


Figure 14 Structure panel for PHP code

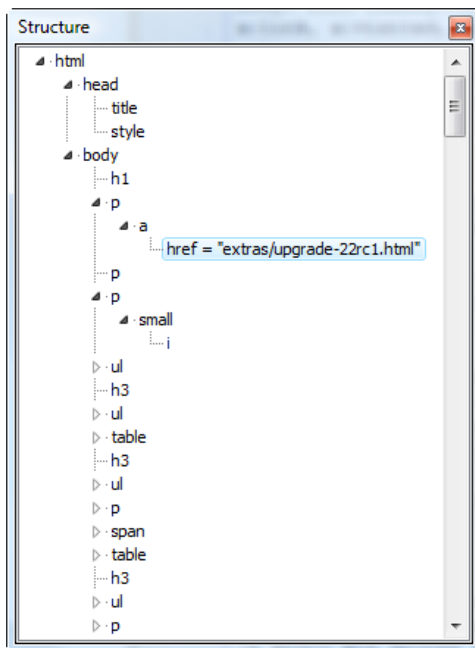


Figure 15 Structure panel for HTML files

THE DEBUGGER

All development runs into problems. When bugs or unexpected application behaviors occur, you need 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.

With Delphi for PHP 2.0, just press F9 and you are running in Debug Mode, place breakpoints on your code and when that code is executed, the IDE allows you to set watches, check the call stack or even profile your application

In Debug mode, the debugger takes over and allows you to control the process of execution and to gain access to all information about the entire process. You can set breakpoints anywhere within the 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 your code. Once execution is halted, the debugger will provide access to all information within scope. Watches can be set 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.

You can also execute without debugging, which is faster and is the preferred method if you don't need any debug capabilities.

BREAKPOINT LIST WINDOW

A breakpoint allows you to stop your program at a specific point and let the IDE take control of your application, so you can check values of variables, call stack, and so on.

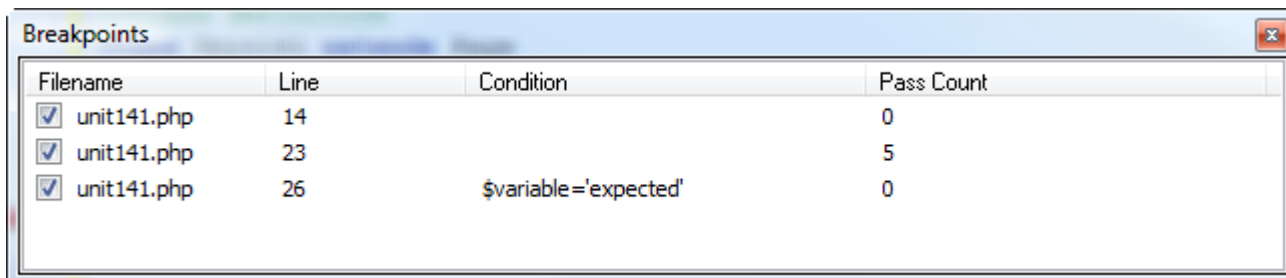


Figure 16 Breakpoints window.

The Breakpoints window lists all breakpoints enabled for your project. A breakpoint can have also a condition, that, when evaluated to true, will enable the breakpoint, and also, a pass count, so the breakpoint will not stop until the program executes certain amount of times that line of code.

You can perform several operations with breakpoints. For example, you can disable them temporarily so they are stored but not working. You can delete them or edit their properties to set conditions and pass counts.

CALL STACK WINDOW

This window is updated at every step in debug mode, listing all function calls that the software has performed, it's very handy to know what the execution flow was that got you to a specific point. If you double-click any entry, the IDE opens the associated source code and points to the function that was called.

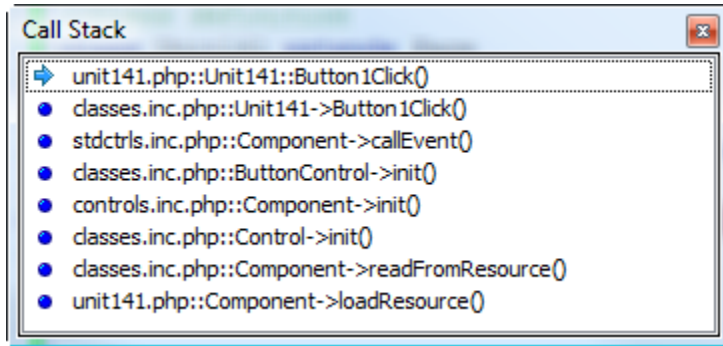


Figure 17 Call Stack Window.

GLOBAL AND 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 you to do just that. As shown in Figure 9, it provides a tree-structured view of the existing global and local variables that are in scope for the given breakpoint. You can examine the current state of both the PHP and the existing request for the page being executed.

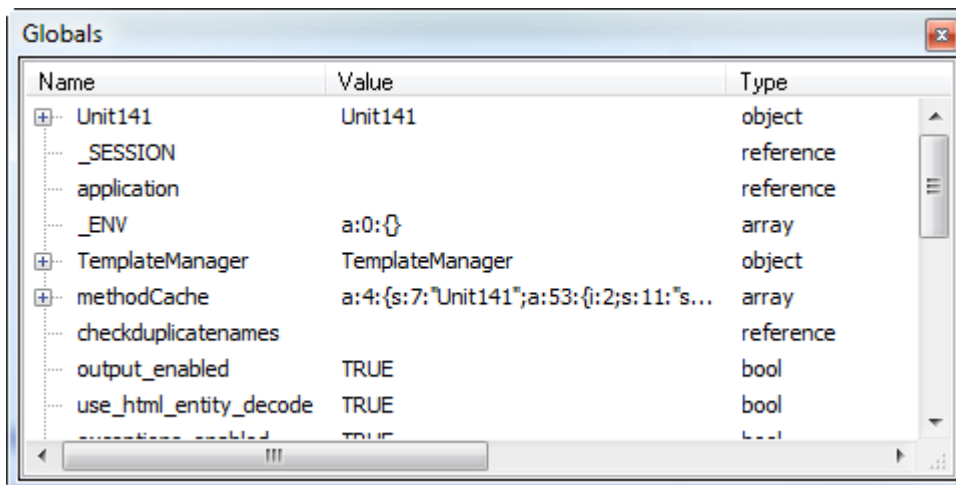


Figure 18 List of Global Variables.

WATCHES WINDOW

This tool allows you to add an expression to be evaluated by the PHP engine at debug time. You can enter simple variables, or even arithmetic operations. The result value is shown at the right, and also the type of the value.

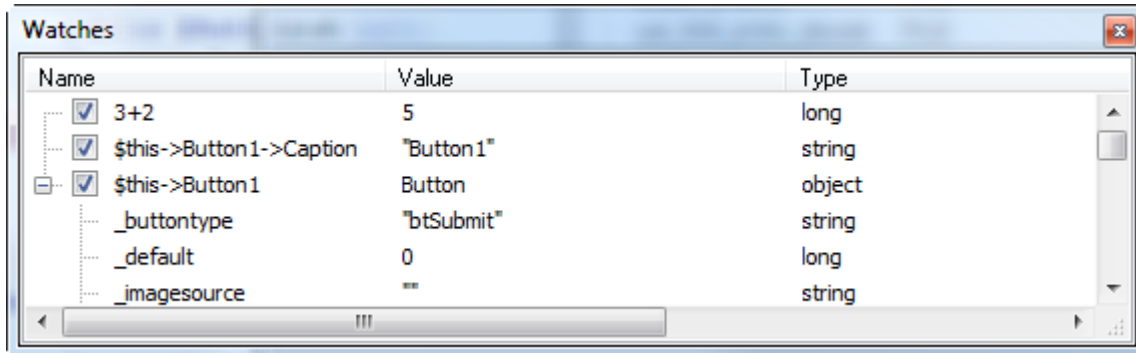


Figure 19 Watches Window.

After enter a watch, you can delete it, disable it, or edit it, to change the expression to be evaluated. The result always depends on the scope of the current execution point.

PROFILER

With this tool, you can get valuable information about the execution of your software, because it lists all lines that have been executed, how many times, the average time spent executing, the total time, and the minimum and maximum execution time. So if there is a process that is running slow or you want to check if you can speed it up, just check the "Enable profiler" checkbox and run your application in Debug mode.

File	Hits	Avg Time (ms)	Total (ms)	Min (ms)	Max (ms)	Path
vd.inc.php(145)	24	20,234	485,631	0,031	156,339	C:\Program Files\CodeGear\Delphi f...
classes.inc.php(921)	959	0,026	24,574	0,017	0,061	C:\Program Files\CodeGear\Delphi f...
classes.inc.php(2240)	909	0,027	24,250	0,019	3,473	C:\Program Files\CodeGear\Delphi f...
system.inc.php(281)	831	0,023	19,461	0,021	0,132	C:\Program Files\CodeGear\Delphi f...
system.inc.php(283)	819	0,019	16,040	0,018	0,063	C:\Program Files\CodeGear\Delphi f...
classes.inc.php(2163)	659	0,022	14,431	0,020	0,053	C:\Program Files\CodeGear\Delphi f...
classes.inc.php(920)	959	0,014	13,467	0,013	0,071	C:\Program Files\CodeGear\Delphi f...
classes.inc.php(2238)	909	0,014	13,042	0,014	0,064	C:\Program Files\CodeGear\Delphi f...
system.inc.php(278)	831	0,014	12,143	0,014	0,051	C:\Program Files\CodeGear\Delphi f...

Figure 20 Profiler.

Once the execution is finished, you will get this report listed in Figure 20, and will be able to check how much time takes to execute specific lines of code. Of course, you can sort by any column to get first the results you are looking for.

CONCLUSION

This article presented the important features from Delphi for PHP considering the PHP developers perspective. With this release, Embarcadero reinforces its commitment with PHP developers, bringing many new features in Delphi for PHP 2 for PHP developers.

Delphi for PHP is the first RAD visual tool for PHP. It allows you to extend and reuse the components through the VCL for PHP component architecture. Additionally, it supports all major databases, and offers a free InterBase Developer Edition for you to use in your development efforts.

The Embarcadero Developer Network (<http://edn.embarcadero.com>) has articles and videos about Delphi for PHP, news articles and videos, access <http://dn.codegear.com> and stay connected.

ABOUT THE AUTHOR

In his role at Embarcadero, Andreano Lanusse is focused on helping make sure products are developed to meet customers' expectations. Responsible for defining market strategies for Latin America, he spends a great deal of time attending developer conferences, tradeshows, and user groups part of his duties as Latin Lead Evangelist, while also visiting strategic customers throughout Latin America. You can read more about Andreano Lanusse on his blog (<http://blogs.codegear.com/andreanolanusse>) and reach him at his e-mail address: andreano.lanusse@embarcadero.com.



Embarcadero Technologies, Inc. is a leading provider of award-winning tools for application developers and database professionals so they can design systems right, build them faster and run them better, regardless of their platform or programming language. Ninety of the Fortune 100 and an active community of more than three million users worldwide rely on Embarcadero products to increase productivity, reduce costs, simplify change management and compliance and accelerate innovation. The company's flagship tools include: Embarcadero® Change Manager™, CodeGear™ RAD Studio, DBArtisan®, Delphi®, ER/Studio®, JBuilder® and Rapid SQL®. Founded in 1993, Embarcadero is headquartered in San Francisco, with offices located around the world. Embarcadero is online at www.embarcadero.com.