DEV2QA

PDT
1.1

Magic Members Support

Writer	Date	Comment	Approved
Roy Ganor	6/23/08	1. Dev2QA	

1. Introduction

1.1 Requirement Rationale

(Parts of this document were inspired by Greg's blog - http://greg.chiaraquartet.net)

One of the trickier feature requests for phpDocumentor has been documenting "magic" object properties and methods. By "magic" I am referring to properties and methods that are created dynamically by PHP 5.0+ user space class methods __get, __set, __isset, __unset and __call.

For instance, this code creates a read-only property named **$foo** and a write-only property named **$bar** and a regular property named **$regular** as well as a magic function **borp()** that accepts two parameters.

```php
<?php
class blah {
   private $_thingy;
   private $_bar;

   function __get($var)    {
      switch ($var) {
         case 'foo' :
            return 45;
             ...
      }
   }

   function __set($var, $val)
   {
      switch ($var) {
         case 'bar' :
            $this->_bar = $val;
            break;
          ...
      }
   }

 function __call($method, $params)
   {
      if ($method == 'borp') {
         if (count($params) == 2) {
            return $params[0] * $params[1];
         }
      }
   }
}
```

In order to document this stuff, we want to add 4 new tags that would only be recognized in a class docblock. They are:

1. **@property** - magic class variables that can be both read and written to
2. **@property-read** - magic class variables that can only be read
3. **@property-write** - magic class variable that can only be written to
4. **@method** - magic class methods

```
/**
 * show off @property, @property-read, @property-write, @method
 *
 * @property mixed $regular regular read/write property
 * @property-read int $foo
 * @property-write blah $bar
 * @method int borp() multiply two integers
 */
class blah
{
        ...
}
```

PDT now supports code completion for magic members (properties and methods) that were declared by these php doc tags.

1.2 New Terminology and Documentation

@property - http://manual.phpdoc.org/HTMLSmartyConverter/PHP/phpDocumentor/tutorial_tags.property.pkg.html

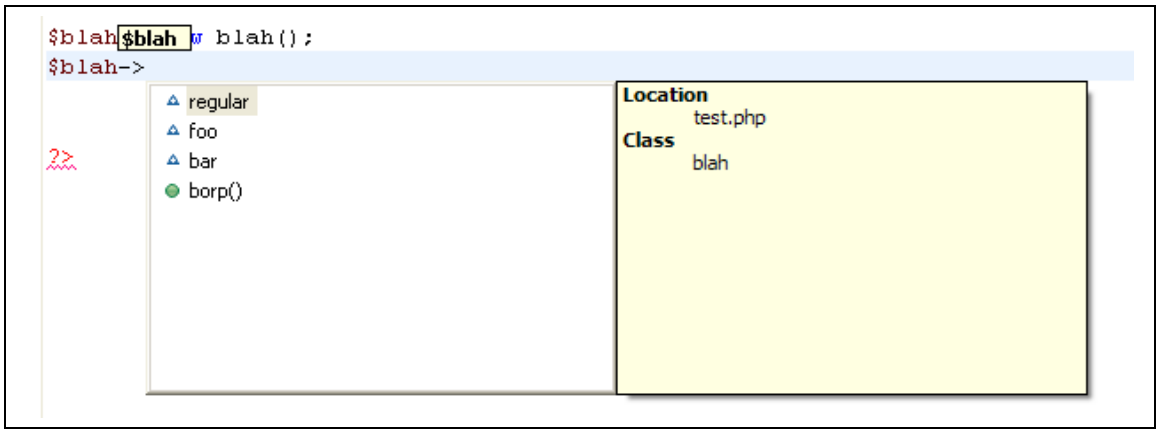
@method – http://manual.phpdoc.org/HTMLSmartyConverter/PHP/phpDocumentor/tutorial_tags.method.pkg.html

2. Detailed Description

2.1 Code completion for magic properties

1. Provide code completion for the declared members.
2. Provide nested code completion for the declared members.
3. Provide hyperlink for magic variables

2.2 Basic UI #1



#	Control Name	Description	Valid Input values	Valid Output Values
1	Editor	Code complete for expressions with type using magic members	$blah->|	regular foo bar borp()
2	Editor	**Nested code complete** for expressions with type using magic members	$blah->bar->|	regular foo bar borp()
3	Editor	**Hyperlink for members (Ctrl + Click)**	$blah->bar = 6 (Ctrl + Click on the bar word)	Should set the caret location to the php doc tag of $bar

3. Unit Testing

This section should be used to define the functionality to be tested at the Unit Test level.

Test Item	**Goal**

Test @property	`@Test`
	`public void testMagicProperty() throws Exception {`
	`  testFile(`
	`  "<? /**\r\n" +`
	`  "  * @property mixed $a\r\n" +`
	`  "  * @property-read int $b\r\n" +`
	`  "  * @property-write array $c\r\n" +`
	`  "  */ " +`
	`  "class A { function foo() { $this->| } } ?>"`
	`  );`
	`}`

Test nesting code completion	`@Test`
	`public void testMagicProperty () throws Exception {`
	`  testFile(`
	`  "<? /**\r\n" +`
	`  "  * @property A $b\r\n" +`
	`  "  */ " +`
	`  "class A { function foo() { }  }" +`
	`  "$a = new A(); $a->b->|  ?>"`
	`  );`
	`}`

4. Future Development

1. Provide argument description for @method tags