```
        $method_name = $method_array[$last_position];
        $record_Array = array(array('dog_name'=>"$this->dog_name", 'dog_weight'=>
        "$this->dog_weight", 'dog_color'=>"$this->dog_color", 'dog_breed'=>"$this->dog_breed"));
        $dog_data->$method_name("Insert",$record_Array);
        $dog_data = NULL;
}
```

The lines to accomplish this are the same as seen previously in the `dog_interface`; you create the `dog_container`, find the location of the `dogdata` file, and create an instance of the `dog_data` class. The only difference is that "dogdata" is passed in for the search. The PHP function `get_class_methods` is used to create a list of methods in the `dog_data` class. The last method in the class is `processRecords`. The name of this method is pulled and placed into `$method_name`. The `record_Array` is then built to be passed into `processRecords`. The method is called, passing `"Insert"` and the `record_Array`. Finally, the `dog_data` object is set to `NULL`, which causes the destructor to save the data.

This allows complete *dependency injection*. The `dog` object does not know the name of the `dog_data` class, the location of the `dog_data` class, or the name of the method to call until it is determined by this code. This creates a complete break between the data tier and the business rules tier, as required for three-tier design.

Example 6-5. The dog.php file using dog_data.php to save data

```
<?php
class Dog
{
// --------------------------------- Properties ----------------------------------------
private $dog_weight = 0;
private $dog_breed = "no breed";
private $dog_color = "no color";
private $dog_name = "no name";
private $error_message = "??";
private $breedxml = "";
// --------------------------------- Constructor ----------------------------------------
function __construct($properties_array)
{
if (method_exists('dog_container', 'create_object')) {
$this->breedxml = $properties_array[4];

$name_error  = $this->set_dog_name($properties_array[0]) == TRUE ? 'TRUE,' : 'FALSE,';
$color_error = $this->set_dog_color($properties_array[2])  == TRUE ? 'TRUE,' : 'FALSE,';
$weight_error= $this->set_dog_weight($properties_array[3]) == TRUE ? 'TRUE' : 'FALSE';
$breed_error = $this->set_dog_breed($properties_array[1])  == TRUE ? 'TRUE,' : 'FALSE,';

$this->error_message = $name_error . $breed_error . $color_error . $weight_error;
$this->save_dog_data();
if(stristr($this->error_message, 'FALSE'))
{
        throw new setException($this->error_message);
} }
else
{ exit; }
}
```