```
<application>
<type ID="breeds">
<location>breeds.xml</location>
</type>
</application>
</dog_applications>
```

In Example 4-10, you have created a simple XML file that will be used for version changes of the significant files in the PHP application. Each `application` tag identifies the type of file (dog, select box, breeds). Each `location` tag within the `application` tag provides the file name and location (although in this example you have all files in the same location). Once you adjust the program to use this file, you will have the flexibility to change the file names (such as the example using `dog3.php` instead of `dog.php`) and the location without having to change any program code. This can help you to swap versions in/out of the application during the development process.

You will now create a `Dog_container` class that will contain two methods. `get_dog_application` will be used to "fetch" the name and location of any of the files listed in the XML file. The `create_object` method will create an instance of either the dog class or the `get_breeds` class.

Example 4-11. The dog_container.php file

```
<?php
class Dog_container
{
private $app;
private $dog_location;
function __construct($value)
{
if (function_exists('clean_input'))
{
$this->app = $value;
}
else
{
exit;
} }
public function set_app($value)
{
$this->app = $value;
}
public function get_dog_application()
{
$xmlDoc = new DOMDocument();
if ( file_exists("dog_applications.xml") )
{
$xmlDoc->load( 'dog_applications.xml' );
$searchNode = $xmlDoc->getElementsByTagName( "type" );
foreach( $searchNode as $searchNode )
{
    $valueID = $searchNode->getAttribute('ID');
    if($valueID == $this->app)
```