

```

public function set_app($value) {
    $this->app = $value; }
public function get_dog_application($search_value) {
    $xmlDoc = new DOMDocument();
    if ( file_exists("e5dog_applications.xml") ) {
        $xmlDoc->load( 'e5dog_applications.xml' );
        $searchNode = $xmlDoc->getElementsByName( "type" );
        foreach( $searchNode as $searchNode ) {
            $valueID = $searchNode->getAttribute('ID');
            if($valueID == $search_value) {
                $xmlLocation = $searchNode->getElementsByName( "location" );
                return $xmlLocation->item(0)->nodeValue;
            }
        }
        throw new Exception("Dog applications xml file missing or corrupt"); }
    }
    function create_object($properties_array) {
        $dog_loc = $this->get_dog_application($this->app);
        if(($dog_loc == FALSE) || (!file_exists($dog_loc))) {
            throw new Exception("File $dog_loc missing or corrupt."); }
        else
        {
            require_once($dog_loc);
            $class_array = get_declared_classes();
            $last_position = count($class_array) - 1;
            $class_name = $class_array[$last_position];
            $dog_object = new $class_name($properties_array);
            return $dog_object;
        }
    }
}
?>

```

The `dog_container` in Example 5-5 replaces returning 'FALSE' from Example 4-10 when the `dog_application.xml` file, `dog.class` file, and/or the `get_breeds` file is missing. Instead, an exception is thrown indicating which file is missing.

**Example 5-6.** The `dog_interface.php` file with exception handling

```

<?php
function clean_input($value)
{
    $value = htmlentities($value);
    $value = strip_tags($value);
    if (get_magic_quotes_gpc())
    {
        $value = stripslashes($value);
    }
    $value = htmlentities($value);
    $bad_chars = array( "{", "}", "(", ")", ";", ":", "<", ">", "/", "$" );
    $value = str_ireplace($bad_chars,"",$value);
    return $value;
}

```