```
    {
    $xmlLocation = $searchNode->getElementsByTagName( "location" );
    return $xmlLocation->item(0)->nodeValue;
    break;
    }
}
}
    return FALSE;
}
function create_object($properties_array)
{
  $dog_loc = $this->get_dog_application();
  if(($dog_loc == FALSE) || (!file_exists($dog_loc)))
  {
    return FALSE;
  }
  else
  {
    require_once($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;
        }
}
}
?>
```

First, at the top of the class (`dog_container`), you declare two private properties—`$app` and `$dog_location`. These properties are declared as private, instead of public, to keep their values only known within this class.

In the constructor, `$value` accepts a name of an application type that you want to find in the XML file (such as `selectbox`). Later in the code, you will compare `$value` to the type ID in the XML file to see if you can find the application type and a file associated with it. The constructor places `$value` into the `$app` property. However, the method also includes an `if` statement that uses the method `function_exists` to determine if the `clean_input` function exists.

Why? At the beginning of the chapter, you briefly looked at code that allowed you to restrict the use of a program to a specific application that has called it. In this example, you look at another technique to restrict which programs can use this class. The `if` statement demands that any program that makes an instance of this class must also have a `clean_input` method. If someone tries to make an instance of the program using another program that does not already contain a `clean_input` method, the `else` part of the statement will execute, which will cause the object not to be created and will close the program.

Security and performance—Any time a program is accepting information from the Internet, or across a network, it is a good idea to determine the source of the information. In addition to determining application and function names, PHP programs can also look at the source IP addresses.