

Encapsulate Fields

Hiding your data and accessing it through an outward interface based on accessor methods is a good idea. Later you can change the data internals, preserving full compatibility with the code relied on the class and its available methods. The *Encapsulate Fields* refactoring allows you to hide your data and create the necessary accessors.

- [Example](#)
- [Performing the Encapsulate Fields refactoring](#)

This refactoring is also available from [UML Class diagram](#).

Example

Before	After
<pre>//File Class.java public class Class { public String aString; }</pre>	<pre>//File Class.java public class Class { private String aString; public void setaString(String aString) { this.aString = aString; } public String getaString() { return aString; } }</pre>
<pre>//File AnotherClass.java public class AnotherClass { public Class aClass; public void method() { aClass.aString="string"; } }</pre>	<pre>//File AnotherClass.java public class AnotherClass { public Class aClass; public void method() { aClass.setaString("string"); } }</pre>

Performing the encapsulate fields refactoring

1. Select a class or a specific field within the class. The way you perform selection depends on the view where you do it.
 - **In the editor:** position the caret at the desired field or at any place inside the class to be refactored.
 - **In the Project view:** select the desired class.
 - **In the Structure view:** select one or more fields.
2. On the main menu or on the context menu of the selection, choose **Refactor | Encapsulate Fields**. The [Encapsulate Fields](#) dialog box is opened, displaying all fields, detected in the selected scope.
3. In the **Fields to encapsulate** area check the fields you want to create accessors for.
4. In the **Encapsulate** area, specify whether you want to create getter or setter methods, or both.
5. To replace all field occurrences with the calls to the appropriate accessor method, in the **Options** area check the option **Use accessor even when field is accessible**.
6. In the **Encapsulated fields' visibility** area specify the new visibility scope for the selected fields.
7. In the **Accessors' visibility** area select the visibility scope for the created accessor methods.
8. [Preview and apply changes](#).

See Also

Reference:

- [Encapsulate Fields Dialog](#)

Web Resources:

- [Developer Community](#) 