

# Encapsulate Fields Dialog

Refactor | Encapsulate Fields

Item	Description
Fields to encapsulate	<p>In this area select the fields you want to create accessors for. You can accept the default method names or change them at will.</p> <p>If a method with the same signature is already present in the class to be refactored, the Method icon  appears to the left of the accessor name, along with the corresponding visibility icon. No new accessor method will be created in this case. The existing method will not be changed and, generally, nothing else will be additionally checked. In this case you have to be attentive whether the existing method actually functions as an accessor. Otherwise, you might need either to choose another name for the accessor or to go back to the editor, change the existing method and only then invoke the Encapsulate Fields refactoring again. If the existing method overrides/implements a parent class' method it will be also marked with the Overrides method  or Implements method  icon.</p> <p>If the accessor method does not exist in the class to be refactored but, if created it would override/implement a method of the parent class such method will be marked with the Overrides method  or Implements method  icon only to the left of the accessor name. A new accessor method will be generated. However, this newly created accessor method will actually override or implement the method of the parent class, that might not exactly be what you planned it for. In this case, you might need either to choose another name for the accessor or to go back to the editor, change the existing method and only then invoke the Encapsulate Fields refactoring again.</p>
Get access/Set access	<p>Use this option group to select which accessor methods (Getter, Setter or both) will be created for the selected fields. If one of the check boxes is cleared, the entire corresponding column (Getter or Setter) in the Fields to Encapsulate table is disabled.</p>
Encapsulated Fields' Visibility	<p>Here you can specify the new visibility scope for the selected fields</p>
Options	<p>Select whether you want to use accessors even when field is accessible or not. If the option <b>Use accessors even when the field is accessible</b> is not checked, the references to the desired fields, when the fields are directly accessible, will not change.</p> <p>Otherwise, all references to the desired fields will be replaced with the accessor calls. It also depends on your selection in the options group <b>Encapsulated Fields' Visibility</b>.</p> <p>For example, if you uncheck the option <b>Use accessors even when the field is accessible</b>, and select the <i>private</i> visibility for the fields, the usages of the fields outside the class will change, but within the class they will remain the same.</p>
Accessors' Visibility	<p>In this area select the visibility scope for the created accessor methods.</p>

## See Also

Procedures:

- [Encapsulate Fields](#)

## Web Resources:

- [Developer Community](#) 