## Menu Option

Menu option exists in correct place with correct text and keyboard shortcut

* Option is Code -> Generate Matcher. Generate Matcher appears in the bottom group of the code menu.
* Shortcut is ctrl-shift-G, ctrl-shift-M

Menu option is enabled when appropriate selection is made. This is when:

* A current project exists and
* A class can be identified, which requires one of:
  + The editor has focus and the caret is within a java class declaration
  + The editor has focus and one or more java class declarations exists in the editor document and one of the declared classes is public
  + The project tool window has focus and a java class is selected
  + The project tool window has focus and a java source file is selected that contains a public class declaration

Menu option is disabled when no appropriate selection is made i.e. the above condition for enabled is not met.

Menu option can be activated by mouse click when enabled and displays the options dialog.

Menu option can be activated by keyboard shortcut when enabled and displays the options dialog.

## Options Dialog

Destination source root defaults to:

If the selected class does not belong to a module (i.e. it is contained in a project dependency rather than the project directly) and the project contains test source roots, the first test source root is chosen

If the selected class does not belong to a module (i.e. it is contained in a project dependency rather than the project directly) and the project contains no test source roots but does contain source roots, the first source root is chosen

If the selected class belongs to a project module and the module contains one or more test source roots and the selected class belongs to one of those test source roots, that test source root is chosen.

If the selected class belongs to a project module and the module contains one or more test source roots and the selected class does not belong to one of those test source roots, the first test source root of the module is chosen.

If the selected class belongs to a project module and the module contains no test source roots and one or more java source roots and the selected class belongs to one of those java source roots (which is necessary I believe), that source root is chosen.

If the selected class belongs to a project module and the module contains no test source roots and one or more java source roots and the selected class does not belong to one of those java source roots (which may not be possible(!)), the first source root of the module is chosen.

If there is no candidate source root, an error dialog with title “No Source Root” displays the message “No source roots have been configured. A source root is required as the target location for the generated class.” The option dialog then closes and no action is taken.

Destination source root drops down to list all java source and test source roots. It does not list resource roots.

Class Name defaults to the name of the selected class with Matcher appended

Destination package defaults to the package name of the selected class

Clicking the … button opens the Choose Destination Package dialog

Selecting a different package and Cancel does not change the selected Destination Package

Selecting a different package and OK changes the selected Destination Package

The selected Destination Package can be edited directly

If the Destination Package does not exist, the text of the part of the path that doesn’t exist is shown in red and the OK button remains enabled.

If the Destination Package is invalid, the invalid part is underlined and the OK button is disabled.

## Generation

If the Destination Package does not exist it is created.

What methods are generated? TODO – if extending another matcher, don’t want to duplicate methods.

Check generator works for single character property names.

Check generator works for property names with initials (URL) etc.

Check generator works for matchers in the default package.

What happens if you try to generate using a superclass that does not have the expected type parameters and constructors?

Generated class should be public

Generated class should be abstract iff option selected for extensible class

Generated class should be final iff option for extensible class is unselected

Generated class should take type parameters R and T (extends source class) iff it is extensible.

Generated class should extend selected superclass if specified, passing R and T if extensible and this type and the source class if not.

Generated class should extend com.mistraltech.smog.core.CompositePropertyMatcher if no superclass specified, passing R and T if extensible and this type and the source class if not.

Generation works for properties with primitive and non-primitive types.