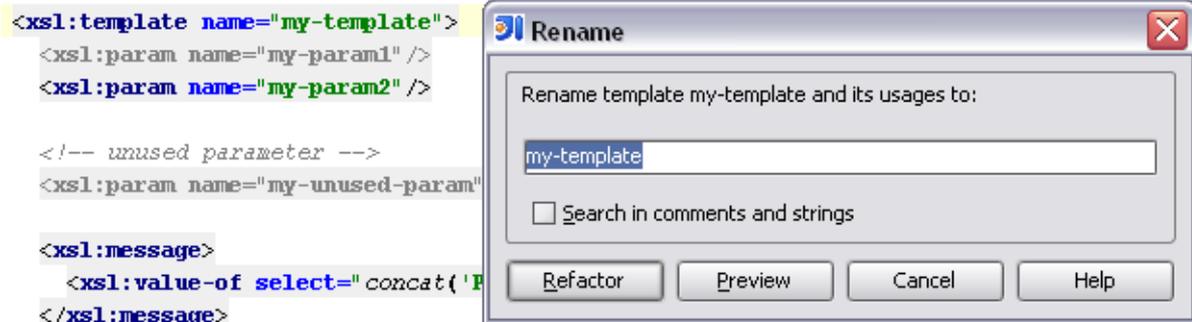


## Refactoring

The plugin supports some of IntelliJ IDEA built-in refactorings **Rename** and **Safe Delete** for XSLT items such as templates, variables and parameters. It's also possible to create XSLT variables from selected XPath expressions with the well-known **Introduce Variable** shortcut. Inlining variables is possible with the **Inline** shortcut as well.

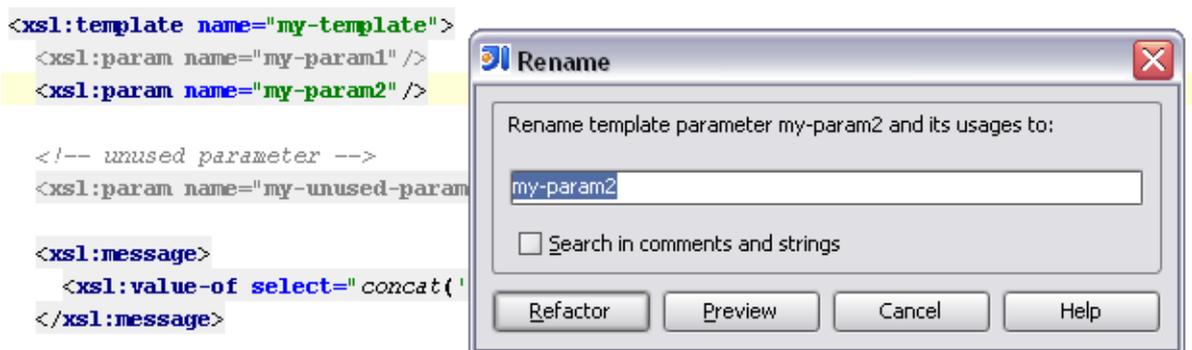
### Renaming templates

Named templates can be renamed in IntelliJ IDEA just like any other symbol. All `xsl:call-template` invocations that refer to this template will be updated accordingly.



### Rename variables and parameters

Just as named templates, it is possible to rename XSLT variables and template parameters, either at a point of their use or at their declaration.



### Safe delete

Named templates, parameters and variables can be deleted using IntelliJ IDEA Safe Delete feature, i.e. the item will be removed if there aren't any references left to it in other stylesheets across the project.

This is especially useful if the stylesheet may be included in other ones via `xsl:include` or `xsl:import` to make sure that nothing will be deleted that is still used somewhere else.

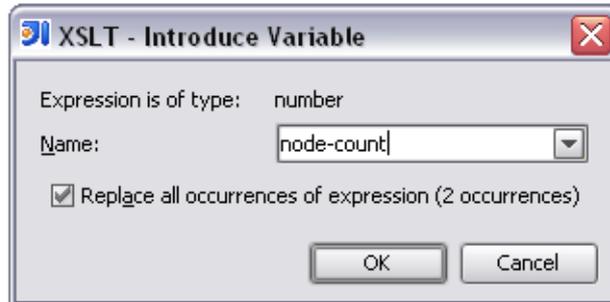
### Introduce variable

It is possible to extract XPath-Expressions and turn them into an `xsl:variable` declaration. Check the **Replace all occurrences** check box to replace all other occurrences of the same expression.

```

<xsl:template match="/">
  <xsl:value-of select="count(//node)" />
  ...
  <xsl:value-of select="count(//node)" />
</xsl:template>

```



### Introduce parameter

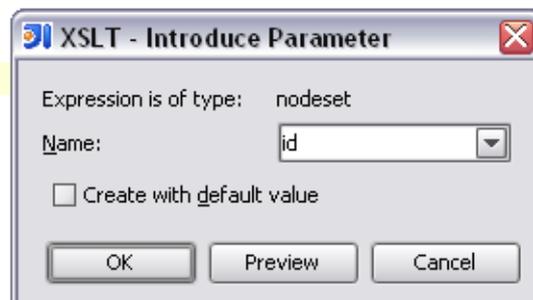
This is similar to **Introduce Variable**, but it creates a new parameter instead of a variable. It also has an additional option **Create with default value** that determines if the selected expression should be added as the introduced parameter's default value or if all calls to the template should be updated to pass the selected expression. That option is only available when introducing parameters to named templates.

```

<xsl:template name="test">
  <xsl:value-of select="foo/@id" />
</xsl:template>

<xsl:template match="/">
  <xsl:call-template name="test" />
</xsl:template>

```



### Inline variable

This is just the opposite of **Introduce Variable**, it replaces all usages of a variable with the expression that is specified in the variable's **select**-attribute. Variables that don't have such an attribute cannot be inlined. Inlining variable references that resolve to parameters is not possible as well.

```

<xsl:variable name="test" select="foo/@id">
  <xsl:value-of select="$test" />

```

