

Logger Insertions

Overview

You can either insert logger statements for the whole class, for a certain method, for a certain variable or at a certain position. These tasks are described in the following subsections.

In the preference page Log4E > User Interaction, you can specify if a certain task should invoke the Preview Wizard. The Preview Wizard shows all differences to the original source code to give you the chance to check where and how logger statements will be created. The modified source can be edited during that process [Pro version only]. If more than one file has been selected you can browse them all in the same wizard and commit the changes at once [Pro version only].

See screenshots on class, position and variable scope for a quick overview. Insertions for method/class

Two operations are executed:

- insert logger statements at certain method entries. These are: method start, method exit, catch block
- surround logger with isLevelEnabled()

Most of the documentation is already done in the Preferences:Positions and Preferences:Statements.

Example:

Before:

```
public String myMethod(String theString, int theInt) {
//Your code....
try {
doSomethingVeryDangerous();
} catch (Exception myexception) {
return null;
}
return toString();
}
```

After:

```
public String myMethod(String theString, int theInt) {
if (logger.isDebugEnabled()) {
logger.debug(
"myMethod(String theString = "
+ theString
+ ", int theInt = "
+ theInt
+ ") - start");
}
//Your code....
try {
doSomethingVeryDangerous();
} catch (Exception myexception) {
logger.error("myMethod()", myexception);
if (logger.isDebugEnabled()) {
logger.debug("myMethod() - end");
}
return null;
}

String returnString = toString();
if (logger.isDebugEnabled()) {
logger.debug("myMethod() - end");
}
return returnString;
}
```

Note that Log4E checks if "returnString" already exists and appends a number. e.g. if "returnString" exists the new variable would be "returnString2" (and so on).

Insert at certain position

Place the textcursor at a valid position within a method and invoke the submenu "Log4E->Insert Log Statement At This Position...". A Wizard pops up as shown below.

Wizard settings:

- Logger message

place your own message there

- Method paramters

List of parameters of this method

- Include paramter type info

If checked the given type of the parameter will be logged too

- Available local variables

List of local variables available at this position

- Include variable type info

If checked the given type of the variable will be logged too

- Level

logger level

- Surround with islevelEnabled()

This depends on the selected logger level.

There's a preview afterwards:

Insertions for selected variable This is an often requested "fast logging feature". It will generate a logger statement immediately without any further interaction. Select any variable within a variable declaration, a variable assignment or a method or place the textcursor below a variable declaration and choose "Log this variable" from the Log4E context menu.