

Exchange of the current logger framework

Overview

The action "Exchange logging framework of this class" automatically changes the actual logging statements of the actual logging framework to the target logging framework. Both logging frameworks has to be defined in the templates (Preferences:Templates).

See the screenshots for a quick overview.

Example:

```
Before:import org.apache.log4j.Logger;public String myMethod(String theString, int theInt) {
if (logger.isDebugEnabled()) {
    logger.debug(
        "myMethod(String theString="
        + theString
        + ", int theInt="
        + theInt
        + ") - my message");
}
}
```

```
//Your code....
try {
    doSomethingVeryDangerous();
} catch (Exception myexception) {
    logger.error("myMethod(String, int)", myexception);
}

return toString();
}
```

```
After exchanging to JDK1.4 logging (logp):import java.util.logging.Level;
import java.util.logging.Logger;
public String myMethod(String theString, int theInt) {
if (logger.isLoggable(Level.CONFIG)) {
    logger.logp(
        Level.CONFIG,
        "MyClass",
        "myMethod(String theString="
        + theString
        + ", int theInt="
        + theInt
        + ")",
        "my message");
}
}
```

```
//Your code....
try {
    doSomethingVeryDangerous();
} catch (Exception myexception) {
    logger.logp(Level.SEVERE, "MyClass", "myMethod(String, int)",
        "", myexception);
}

return toString();
}
```