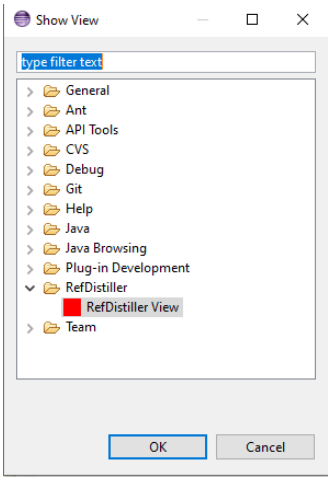
Tool analysis document

Installation and Setup Process:

Initially, there was difficulty in downloading RefDistiller due to the original links being outdated and not maintained. However, the required page was eventually found in an archived web collection. Despite this hiccup, the installation process proceeded smoothly once the correct resource was obtained. RefDistiller was installed as a standard plugin in the Eclipse IDE, with the necessary jar files added inside the plugins folder of the Eclipse IDE installation location, from a rar package found on the plugin's website.

Usage Experience:



RefDistiller was used to compare different versions of a program, featuring an intuitive interface for selecting program versions. It employs two main techniques: RefChecker, for detecting missing edits, and RefSeparator, for identifying additional edits that could alter the program's behavior. The tool supports six common Java program refactoring types, offering detailed insights into potential issues, missing edits, and extra edits. This allows for a focused and efficient review process by pinpointing specific deviations from intended refactoring edits. The tool can be accessed by selecting Window-> Show View-> Other leading to the screen showed on the right.

Benefits and Findings:

The primary advantage of RefDistiller is its ability to detect subtle refactoring errors that could lead to significant changes in application behavior, errors that are often missed during manual refactoring. It serves not only as a tool for identifying these errors but also as an educational resource on the importance of meticulous refactoring practices. In a test project, RefDistiller highlighted several potential deviations and extra edits, illustrating the complexity of manual refactoring and the ease of introducing errors.