Noopur Gupta

Eclipse JDT/UI Committer

IBM India

noopur_gupta@in.ibm.com

Eclipse provides a lot of powerful features and capabilities, which are not easily discoverable and are not leveraged to the fullest.

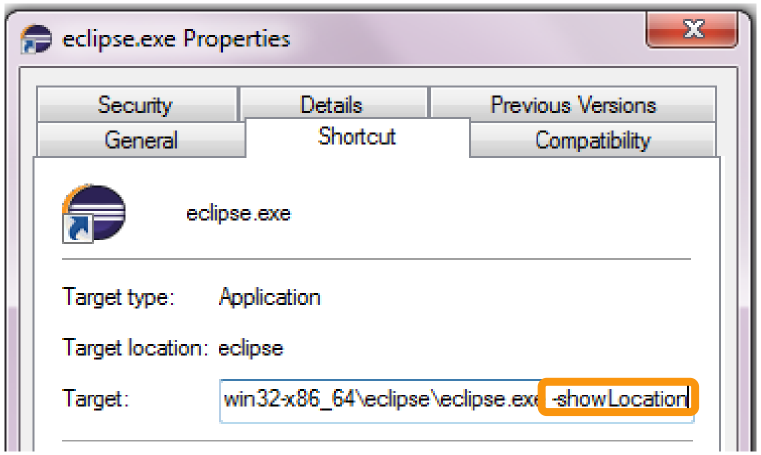
To be productive, mastering your IDE is as important as mastering your source code.

Organizing

- Multiple Workspaces
- Projects in a Workspace
- Inside a Project
- Share Preferences between Workspaces

Multiple Workspaces

- Show Workspace Location in the Title Bar *(-showLocation)*



eclipse.exe Properties

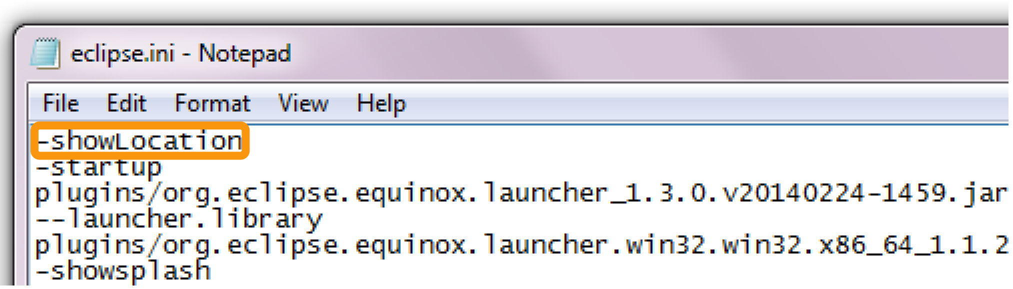
eclipse.exe

Target type: Application

Target location: eclipse

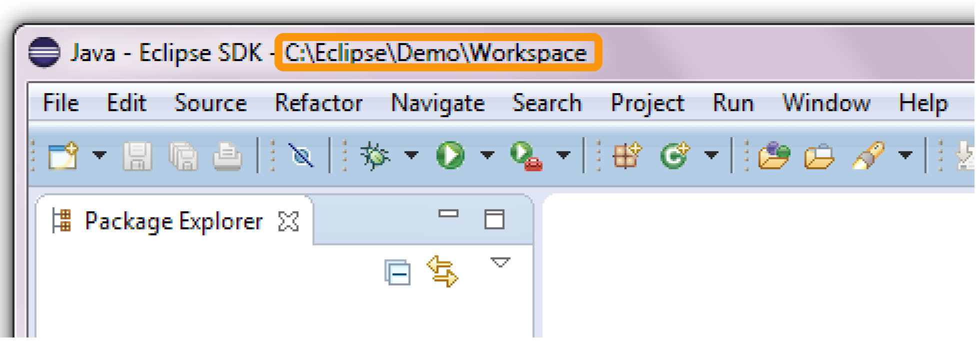
Target: win32-x86_64\eclipse\eclipse.exe -showLocation

OR



eclipse.ini - Notepad

```
-showLocation
-startup
plugins/org.eclipse.equinox.launcher_1.3.0.v20140224-1459.jar
--launcher.library
plugins/org.eclipse.equinox.launcher.win32.win32.x86_64_1.1.2
-showsplash
```



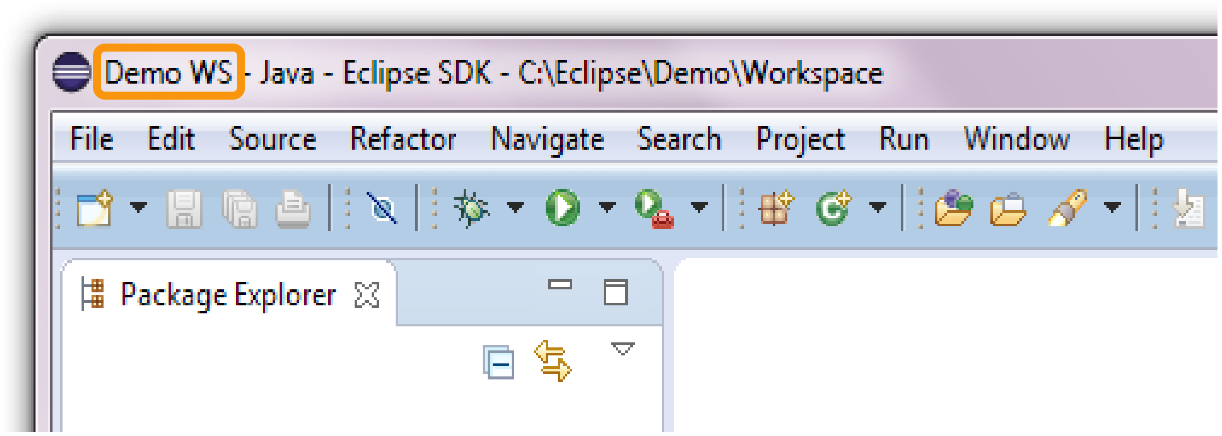
Java - Eclipse SDK - C:\Eclipse\Demo\Workspace

File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer

Multiple Workspaces

- Show Workspace Name in the Title Bar

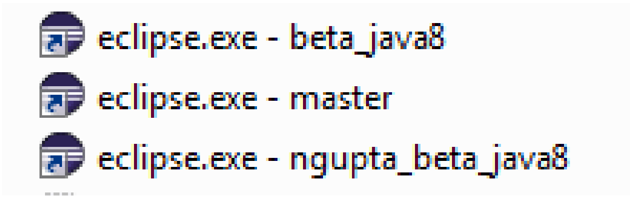
(Window > Preferences > General > Workspace)

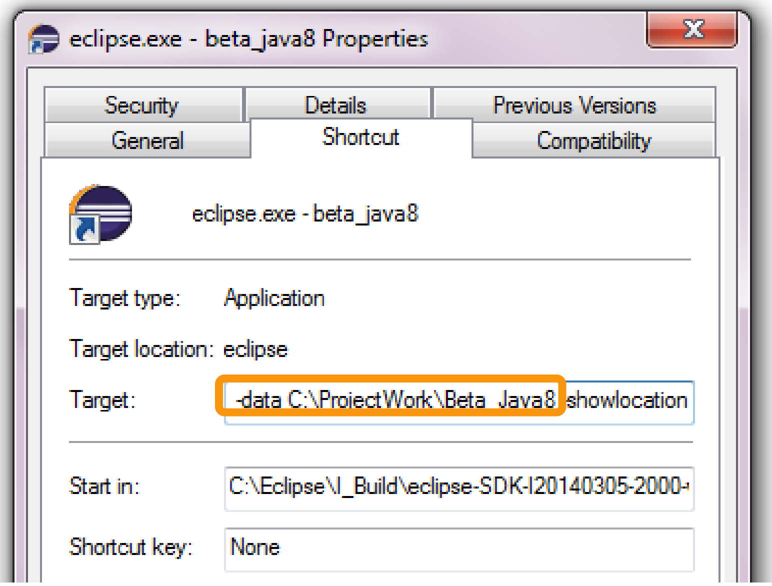

Demo WS - Java - Eclipse SDK - C:\Eclipse\Demo\Workspace
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer

Multiple Workspaces

- Create Eclipse shortcuts with default workspaces

(-data workspacePath)


eclipse.exe - beta_java8
eclipse.exe - master
eclipse.exe - ngupta_beta_java8


eclipse.exe - beta_java8 Properties

eclipse.exe - beta_java8

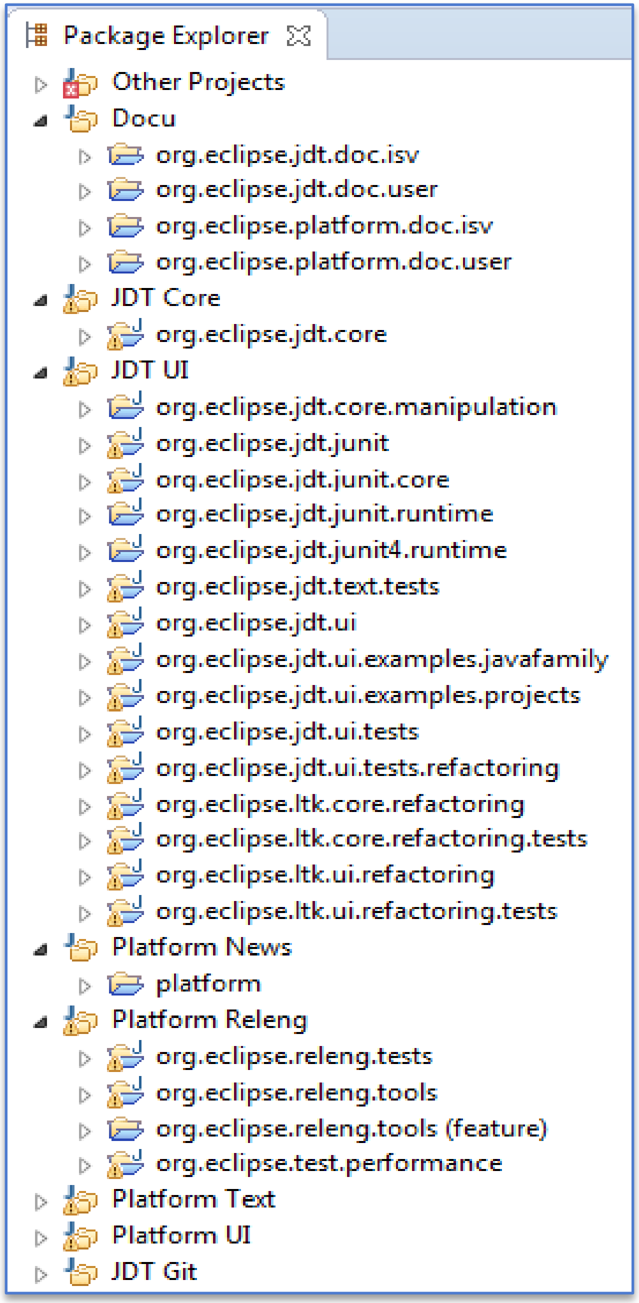
Target type: Application
Target location: eclipse
Target: -data C:\ProjectWork\Beta_Java8 showlocation
Start in: C:\Eclipse\I_Build\eclipse-SDK-I20140305-2000-
Shortcut key: None

Projects in a Workspace

- Working Sets

Package Explorer >
Configure Working Sets…

Package Explorer >
Top Level Elements >
Working Sets

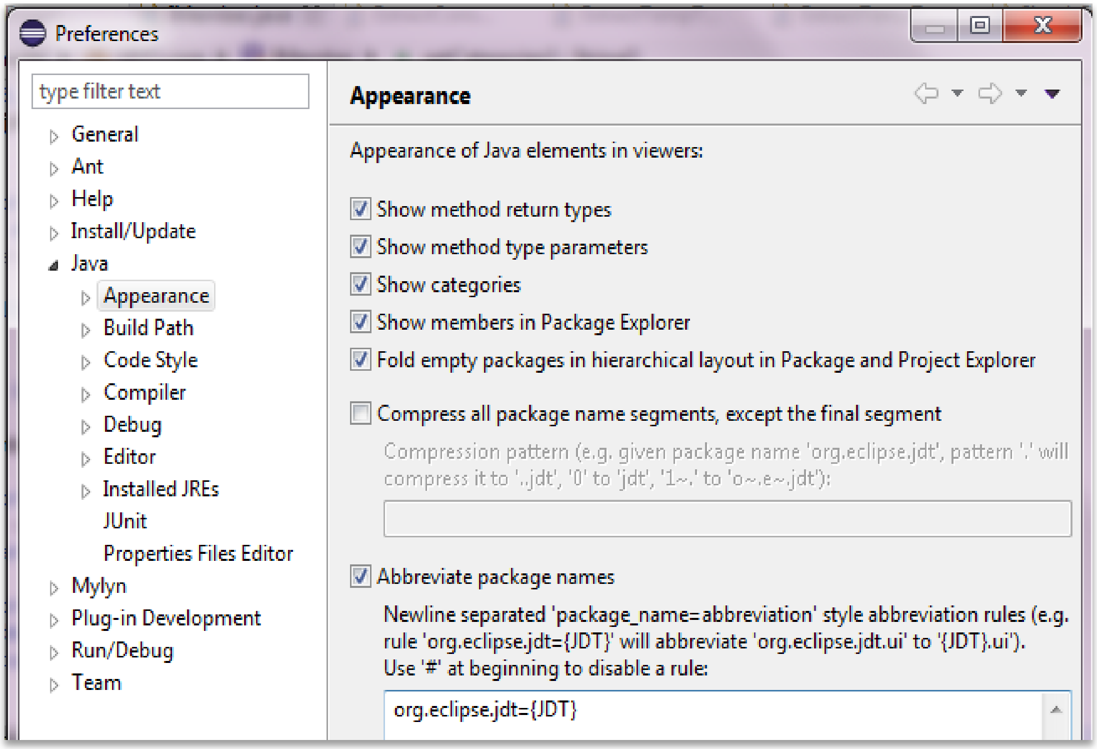

Package Explorer

- Other Projects
- Docu
  - org.eclipse.jdt.doc.isv
  - org.eclipse.jdt.doc.user
  - org.eclipse.platform.doc.isv
  - org.eclipse.platform.doc.user
- JDT Core
  - org.eclipse.jdt.core
- JDT UI
  - org.eclipse.jdt.core.manipulation
  - org.eclipse.jdt.junit
  - org.eclipse.jdt.junit.core
  - org.eclipse.jdt.junit.runtime
  - org.eclipse.jdt.junit4.runtime
  - org.eclipse.jdt.text.tests
  - org.eclipse.jdt.ui
  - org.eclipse.jdt.ui.examples.javafamily
  - org.eclipse.jdt.ui.examples.projects
  - org.eclipse.jdt.ui.tests
  - org.eclipse.jdt.ui.tests.refactoring
  - org.eclipse.ltk.core.refactoring
  - org.eclipse.ltk.core.refactoring.tests
  - org.eclipse.ltk.ui.refactoring
  - org.eclipse.ltk.ui.refactoring.tests
- Platform News
  - platform
- Platform Releng
  - org.eclipse.releng.tests
  - org.eclipse.releng.tools
  - org.eclipse.releng.tools (feature)
  - org.eclipse.test.performance
- Platform Text
- Platform UI
- JDT Git

Inside a Project

- Abbreviate package names with custom rules

Window > Preferences >
Java > Appearance >
Abbreviate package names



Appearance of Java elements in viewers:

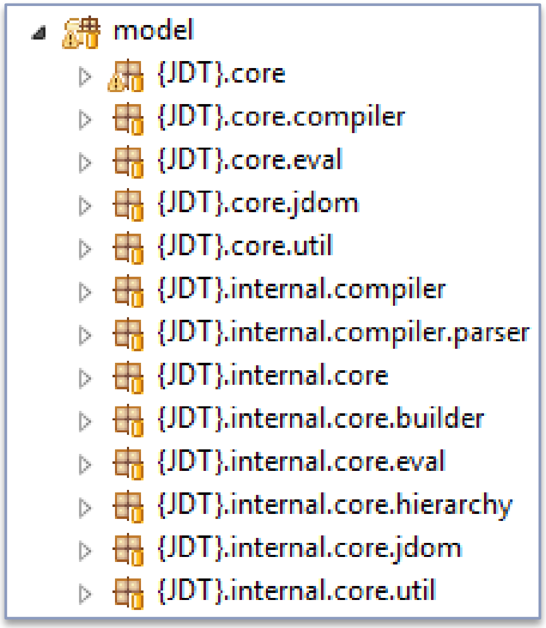
- [✓] Show method return types
- [✓] Show method type parameters
- [✓] Show categories
- [✓] Show members in Package Explorer
- [✓] Fold empty packages in hierarchical layout in Package and Project Explorer
- [ ] Compress all package name segments, except the final segment

Compression pattern (e.g. given package name 'org.eclipse.jdt', pattern '.' will compress it to '..jdt', '0' to 'jdt', '1~.' to 'o~.e~.jdt'):

- [✓] Abbreviate package names

Newline separated 'package_name=abbreviation' style abbreviation rules (e.g. rule 'org.eclipse.jdt={JDT}' will abbreviate 'org.eclipse.jdt.ui' to '{JDT}.ui'). Use '#' at beginning to disable a rule:

```
org.eclipse.jdt={JDT}
```



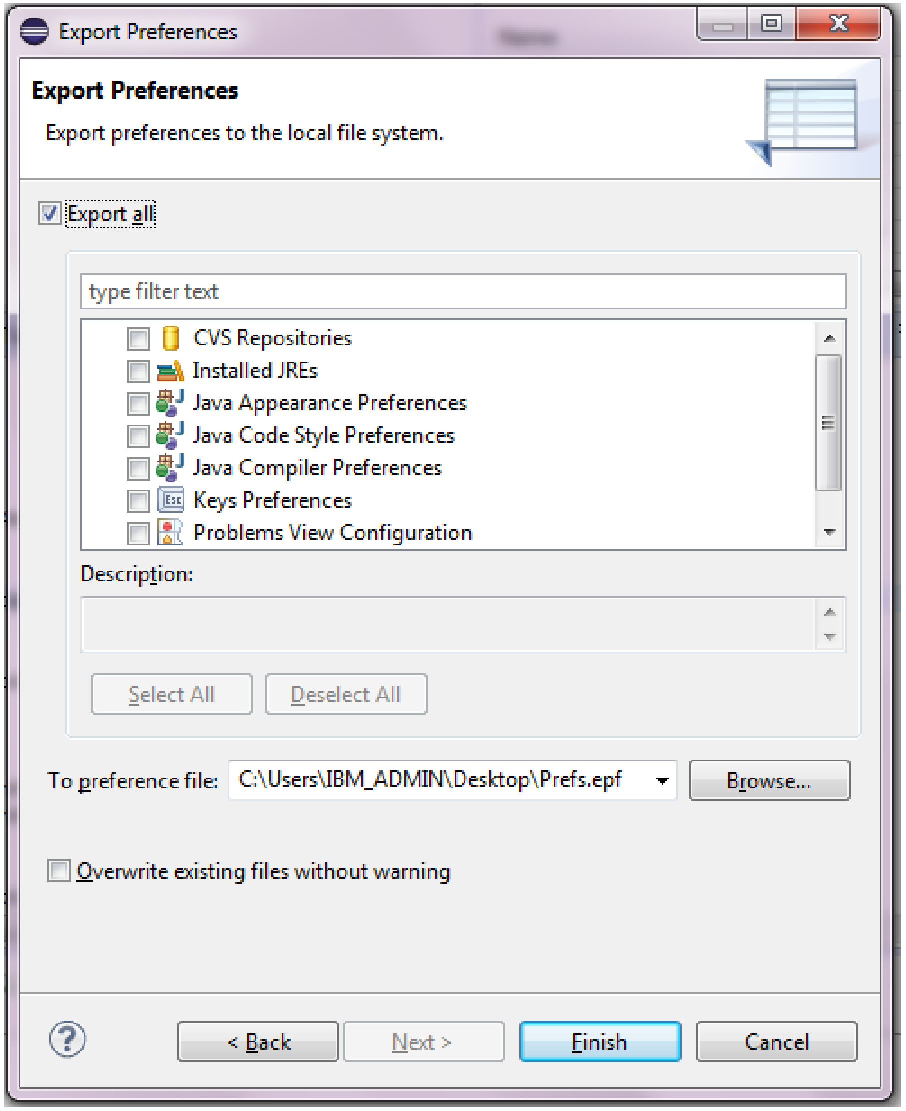
```
model
  {JDT}.core
  {JDT}.core.compiler
  {JDT}.core.eval
  {JDT}.core.jdom
  {JDT}.core.util
  {JDT}.internal.compiler
  {JDT}.internal.compiler.parser
  {JDT}.internal.core
  {JDT}.internal.core.builder
  {JDT}.internal.core.eval
  {JDT}.internal.core.hierarchy
  {JDT}.internal.core.jdom
  {JDT}.internal.core.util
```

Share Preferences between Workspaces

- Export/Import

File > Export…
General > Preferences

File > Import…
General > Preferences


Export Preferences

Export Preferences

Export preferences to the local file system.

[✓] Export all

type filter text

- [ ] CVS Repositories
- [ ] Installed JREs
- [ ] Java Appearance Preferences
- [ ] Java Code Style Preferences
- [ ] Java Compiler Preferences
- [ ] Keys Preferences
- [ ] Problems View Configuration

Description:

Select All | Deselect All

To preference file: C:\Users\IBM_ADMIN\Desktop\Prefs.epf | Browse...

[ ] Overwrite existing files without warning

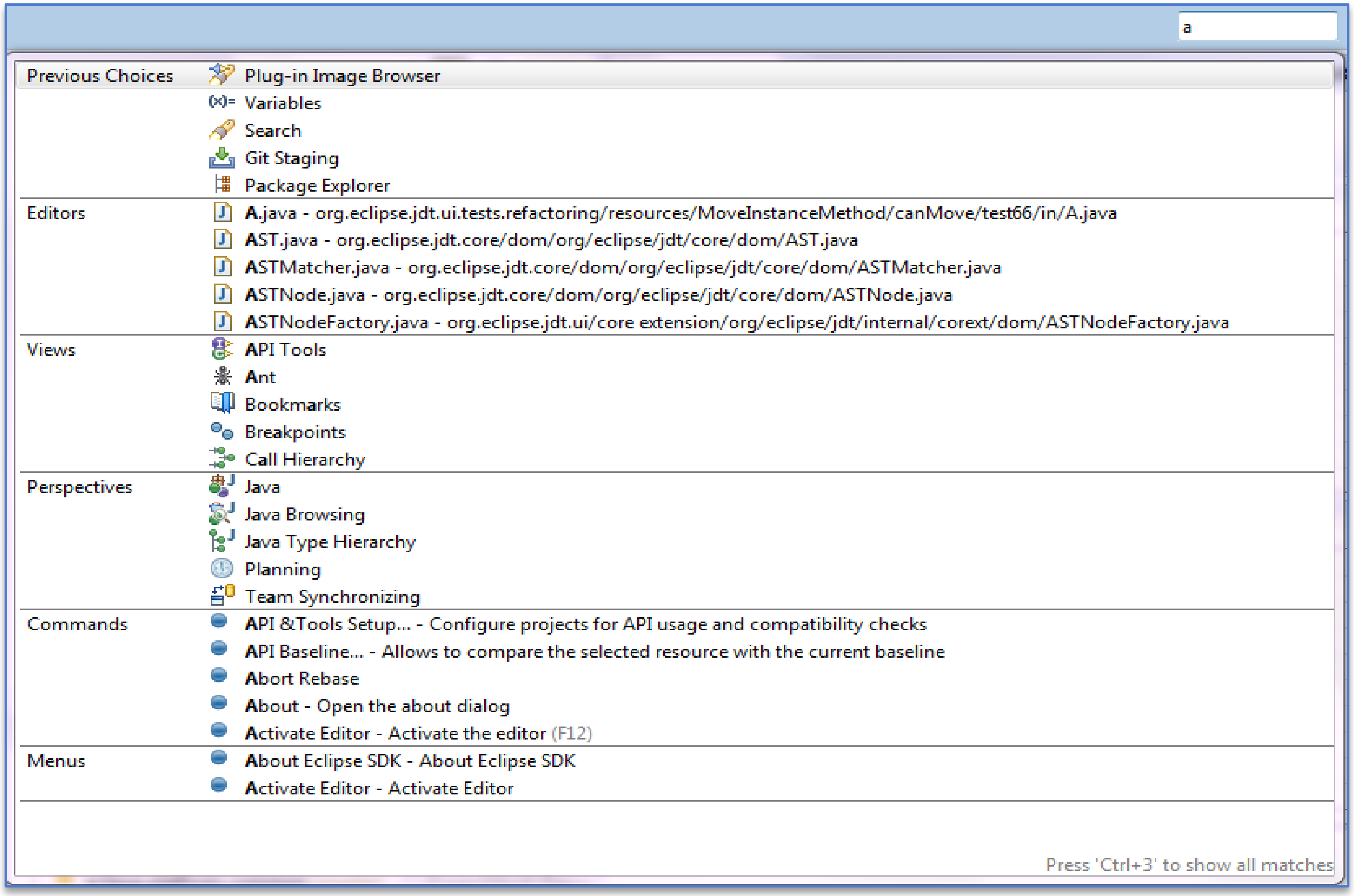
< Back | Next > | Finish | Cancel

Navigating

- Quick Access
- Quick Outline, In-place Outline
- Breadcrumb
- Ctrl+Click for Externalized Strings
- Find Problems with Externalized Strings
- Java Stack Trace Console

Quick Access Ctrl + 3

- **Talk to Eclipse :** Start typing and get the results from many categories of UI elements.



a

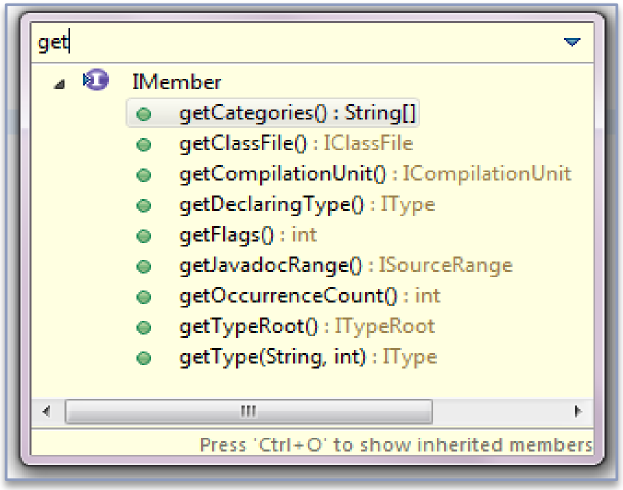
Previous Choices	Plug-in Image Browser
	Variables
	Search
	Git Staging
	Package Explorer
Editors	A.java - org.eclipse.jdt.ui.tests.refactoring/resources/MoveInstanceMethod/canMove/test66/in/A.java
	AST.java - org.eclipse.jdt.core/dom/org/eclipse/jdt/core/dom/AST.java
	ASTMatcher.java - org.eclipse.jdt.core/dom/org/eclipse/jdt/core/dom/ASTMatcher.java
	ASTNode.java - org.eclipse.jdt.core/dom/org/eclipse/jdt/core/dom/ASTNode.java
	ASTNodeFactory.java - org.eclipse.jdt.ui/core extension/org/eclipse/jdt/internal/corext/dom/ASTNodeFactory.java
Views	API Tools
	Ant
	Bookmarks
	Breakpoints
	Call Hierarchy
Perspectives	Java
	Java Browsing
	Java Type Hierarchy
	Planning
	Team Synchronizing
Commands	API &Tools Setup... - Configure projects for API usage and compatibility checks
	API Baseline... - Allows to compare the selected resource with the current baseline
	Abort Rebase
	About - Open the about dialog
	Activate Editor - Activate the editor (F12)
Menus	About Eclipse SDK - About Eclipse SDK
	Activate Editor - Activate Editor

Press 'Ctrl+3' to show all matches

Quick Outline and In-place Outline

- **Quick Outline:** Ctrl + O

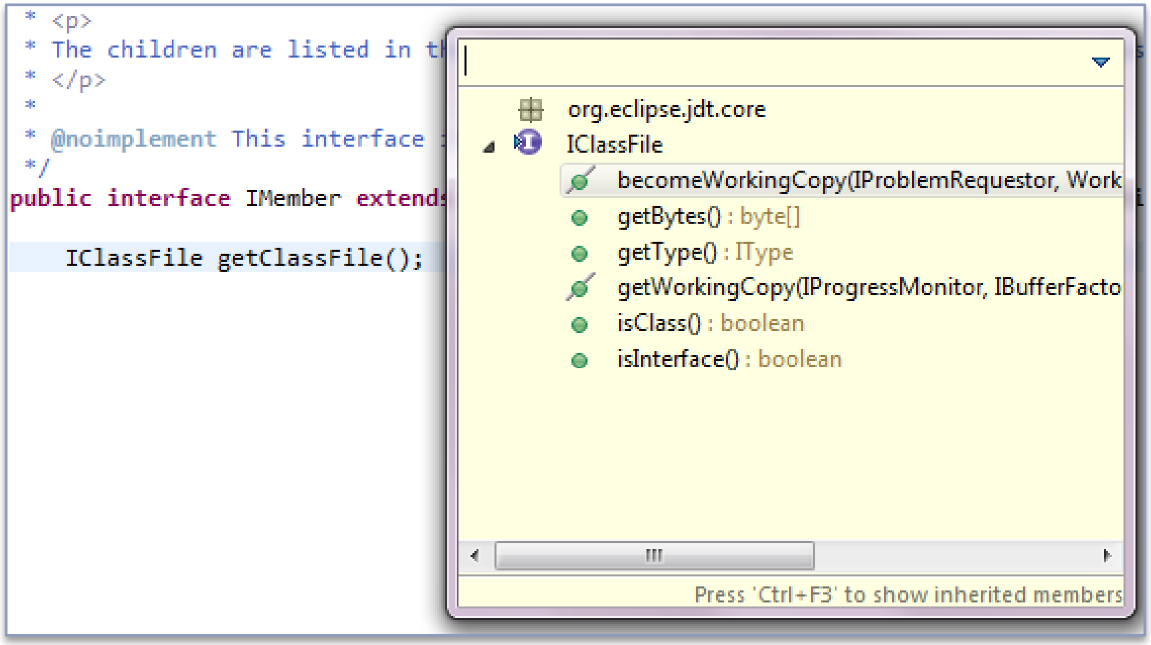
To list the structural elements of the file (such as classes, fields, methods for a Java source file).



```
get
IMember
  getCategories() : String[]
  getClassFile() : IClassFile
  getCompilationUnit() : ICompilationUnit
  getDeclaringType() : IType
  getFlags() : int
  getJavadocRange() : ISourceRange
  getOccurrenceCount() : int
  getTypeRoot() : ITypeRoot
  getType(String, int) : IType
Press 'Ctrl+O' to show inherited members
```

- **In-place Outline:** Ctrl + F3

To pop up an in-place outline of the element at the current cursor position.



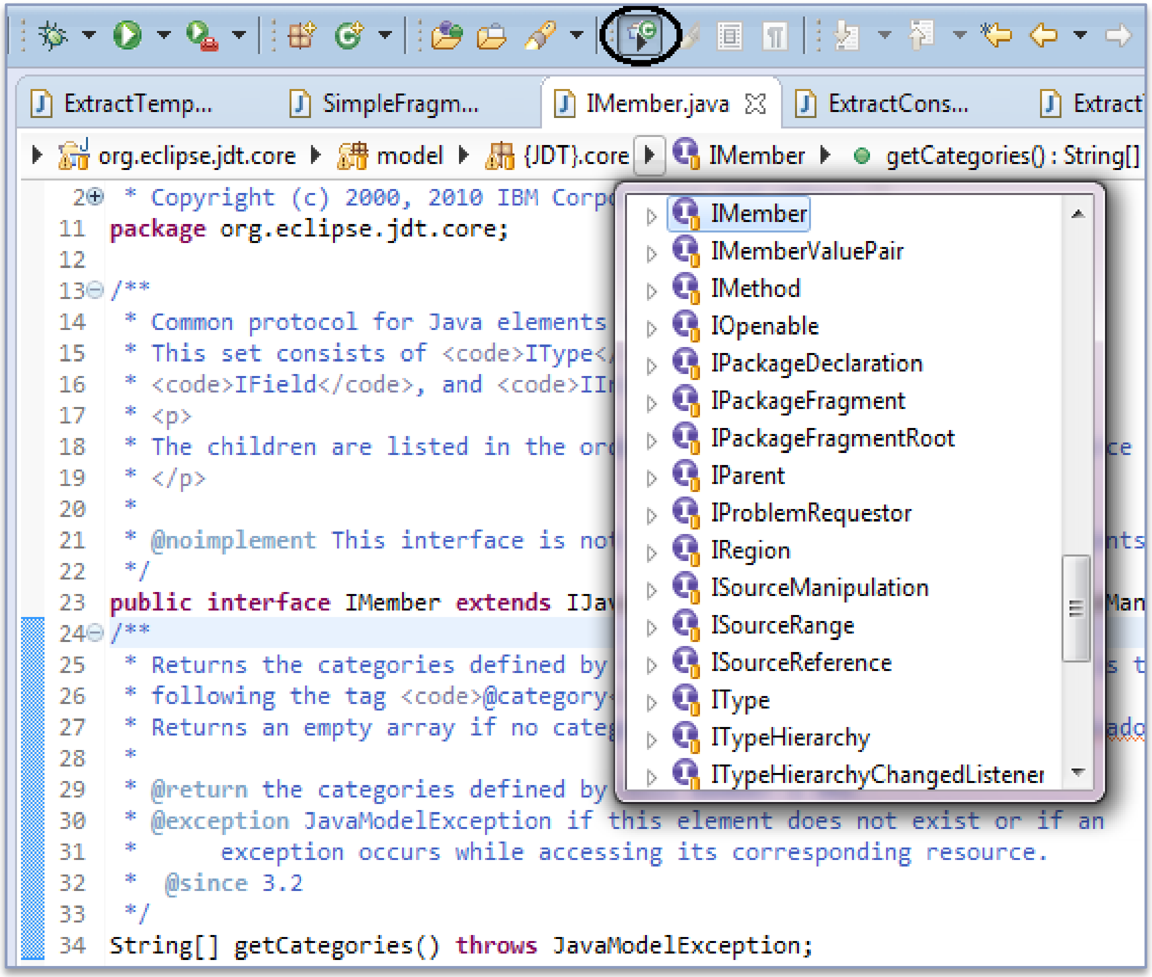
```
 * <p>
 * The children are listed in th
 * </p>
 *
 * @noimplement This interface i
 */
public interface IMember extends

    IClassFile getClassFile();
```

```
org.eclipse.jdt.core
IClassFile
  becomeWorkingCopy(IProblemRequestor, Work
  getBytes() : byte[]
  getType() : IType
  getWorkingCopy(IProgressMonitor, IBufferFacto
  isClass() : boolean
  isInterface() : boolean
Press 'Ctrl+F3' to show inherited members
```

Java Editor Breadcrumb

***Toggle Breadcrumb** tool bar button*

- Shows the path to the element at the cursor position.
- Navigate to other elements via drop-downs and invoke actions (when other views are not visible).
- Also available on multiple editors that are open side-by-side.



ExtractTemp... | SimpleFragm... | IMember.java | ExtractCons... | Extract

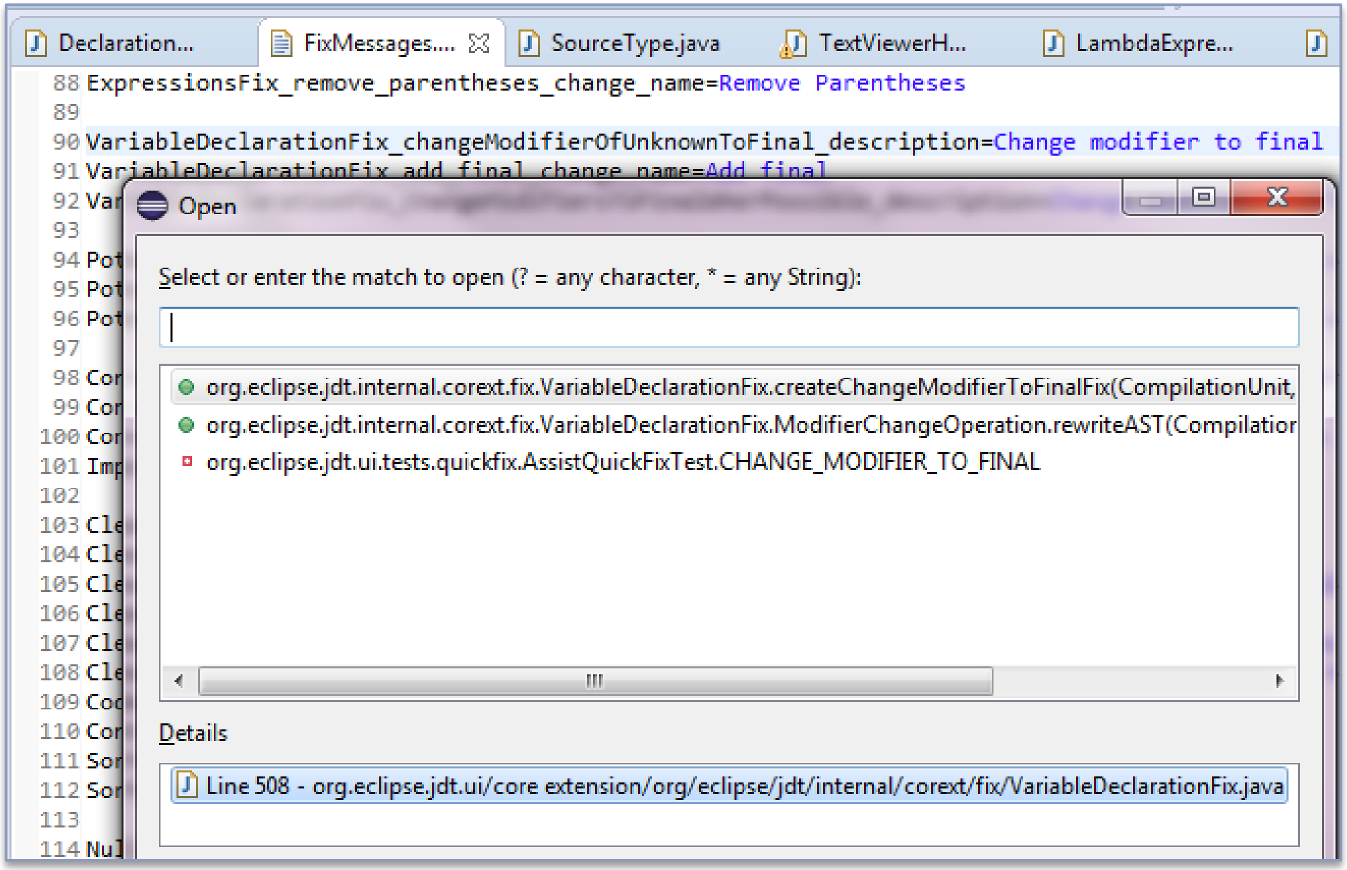
org.eclipse.jdt.core ▸ model ▸ {JDT}.core ▸ IMember ▸ getCategories() : String[]

```
 * Copyright (c) 2000, 2010 IBM Corp
package org.eclipse.jdt.core;

/**
 * Common protocol for Java elements
 * This set consists of <code>IType</
 * <code>IField</code>, and <code>III
 * <p>
 * The children are listed in the or                              ce
 * </p>
 *
 * @noimplement This interface is no                              nts
 */
public interface IMember extends IJa                               Man
/**
 * Returns the categories defined by                               s t
 * following the tag <code>@category
 * Returns an empty array if no categ                              ado
 *
 * @return the categories defined by
 * @exception JavaModelException if this element does not exist or if an
 *      exception occurs while accessing its corresponding resource.
 *  @since 3.2
 */
String[] getCategories() throws JavaModelException;
```

- IMember
- IMemberValuePair
- IMethod
- IOpenable
- IPackageDeclaration
- IPackageFragment
- IPackageFragmentRoot
- IParent
- IProblemRequestor
- IRegion
- ISourceManipulation
- ISourceRange
- ISourceReference
- IType
- ITypeHierarchy
- ITypeHierarchyChangedListener

Ctrl+Click for Externalized Strings

- Ctrl+Click on a property key in the *.properties file shows the places in source code where it is being used and takes you to the referencing code.



```
88 ExpressionsFix_remove_parentheses_change_name=Remove Parentheses
89
90 VariableDeclarationFix_changeModifierOfUnknownToFinal_description=Change modifier to final
91 VariableDeclarationFix_add_final_change_name=Add final
92 Var
93
94 Pot
95 Pot
96 Pot
97
98 Cor
99 Cor
100 Cor
101 Imp
102
103 Cle
104 Cle
105 Cle
106 Cle
107 Cle
108 Cle
109 Coo
110 Cor
111 Sor
112 Sor
113
114 Nul
```

Open

Select or enter the match to open (? = any character, * = any String):

- org.eclipse.jdt.internal.corext.fix.VariableDeclarationFix.createChangeModifierToFinalFix(CompilationUnit,
- org.eclipse.jdt.internal.corext.fix.VariableDeclarationFix.ModifierChangeOperation.rewriteAST(Compilatior
- org.eclipse.jdt.ui.tests.quickfix.AssistQuickFixTest.CHANGE_MODIFIER_TO_FINAL

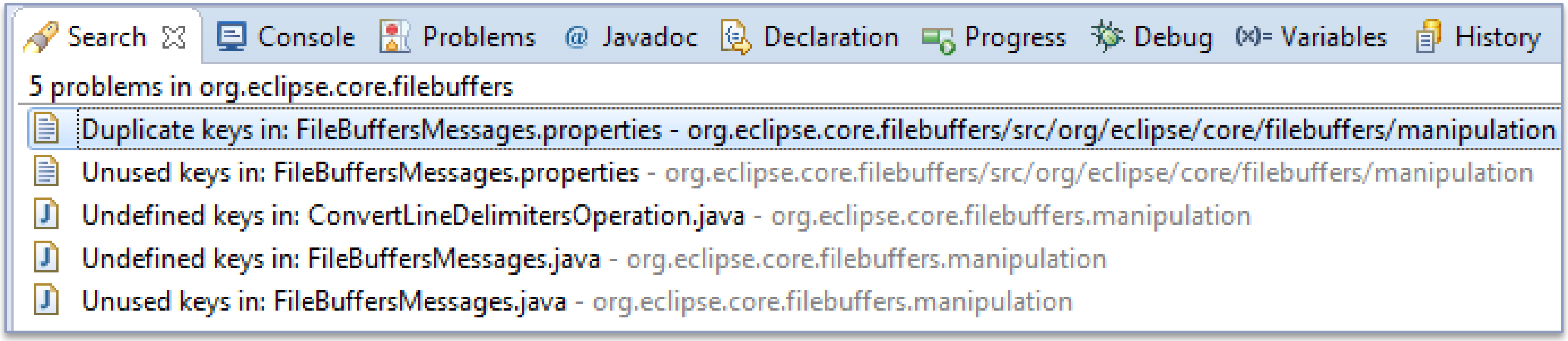
Details

Line 508 - org.eclipse.jdt.ui/core extension/org/eclipse/jdt/internal/corext/fix/VariableDeclarationFix.java

Find Problems with Externalized Strings

Source > Find Broken Externalized Strings

- Finds undefined, unused and duplicate keys.



```
5 problems in org.eclipse.core.filebuffers
Duplicate keys in: FileBuffersMessages.properties - org.eclipse.core.filebuffers/src/org/eclipse/core/filebuffers/manipulation
Unused keys in: FileBuffersMessages.properties - org.eclipse.core.filebuffers/src/org/eclipse/core/filebuffers/manipulation
Undefined keys in: ConvertLineDelimitersOperation.java - org.eclipse.core.filebuffers.manipulation
Undefined keys in: FileBuffersMessages.java - org.eclipse.core.filebuffers.manipulation
Unused keys in: FileBuffersMessages.java - org.eclipse.core.filebuffers.manipulation
```

Java Stack Trace Console

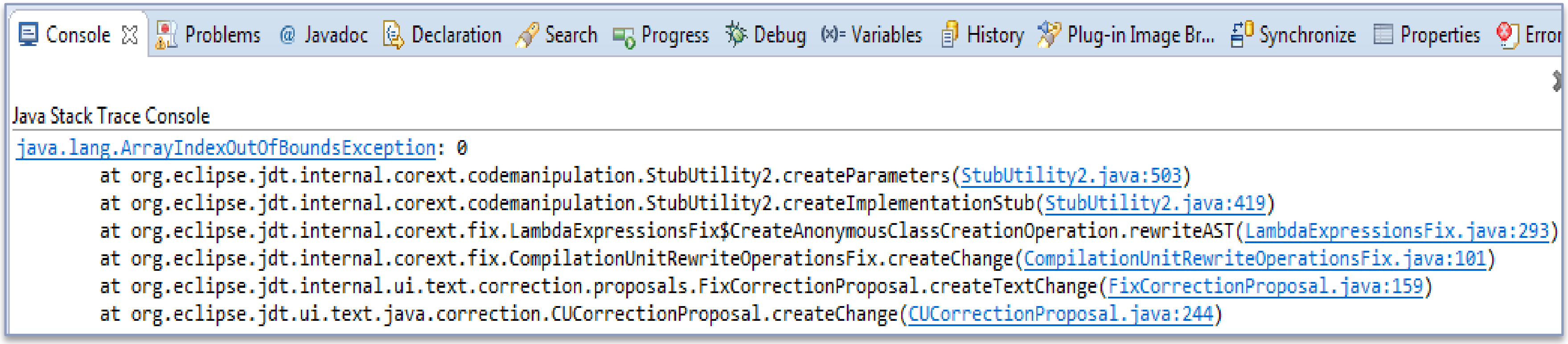
- From a stack trace in log file, instead of locating the file and going to the line number via ***Ctrl+L***, use Java Stack Trace Console.
- Copy the stack trace from log file and click:

Navigate > Open from Clipboard   OR

- Copy the stack trace and paste on the console:

Console view > Open Console (drop-down menu) > Java Stack Trace Console

- Click on the hyperlinks for Java class names with line numbers to navigate.
- Clicking on the exception name in stack trace will create an exception breakpoint.

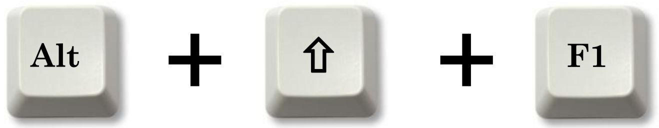


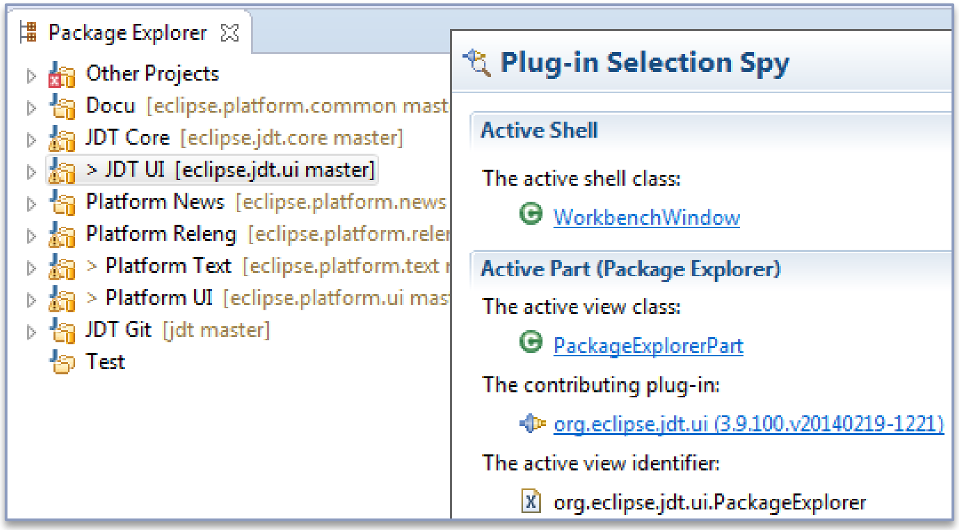
Java Stack Trace Console

```
java.lang.ArrayIndexOutOfBoundsException: 0
	at org.eclipse.jdt.internal.corext.codemanipulation.StubUtility2.createParameters(StubUtility2.java:503)
	at org.eclipse.jdt.internal.corext.codemanipulation.StubUtility2.createImplementationStub(StubUtility2.java:419)
	at org.eclipse.jdt.internal.corext.fix.LambdaExpressionsFix$CreateAnonymousClassCreationOperation.rewriteAST(LambdaExpressionsFix.java:293)
	at org.eclipse.jdt.internal.corext.fix.CompilationUnitRewriteOperationsFix.createChange(CompilationUnitRewriteOperationsFix.java:101)
	at org.eclipse.jdt.internal.ui.text.correction.proposals.FixCorrectionProposal.createTextChange(FixCorrectionProposal.java:159)
	at org.eclipse.jdt.ui.text.java.correction.CUCorrectionProposal.createChange(CUCorrectionProposal.java:244)
```

- **Coding**
  - Spy
  - Paste code on Package Explorer
  - Show Annotations
  - Hover
  - Type Filters
  - Templates
  - Search menu actions
  - Block Selection
  - Formatter Off/On Tags
  - Quick Fixes and Quick Assists
  - Keyboard Shortcuts

Plug-in Spy




Package Explorer
Other Projects
Docu [eclipse.platform.common mast
JDT Core [eclipse.jdt.core master]
> JDT UI [eclipse.jdt.ui master]
Platform News [eclipse.platform.news
Platform Releng [eclipse.platform.relen
> Platform Text [eclipse.platform.text r
> Platform UI [eclipse.platform.ui mas
JDT Git [jdt master]
Test

Plug-in Selection Spy

Active Shell

The active shell class:

WorkbenchWindow

Active Part (Package Explorer)

The active view class:

PackageExplorerPart

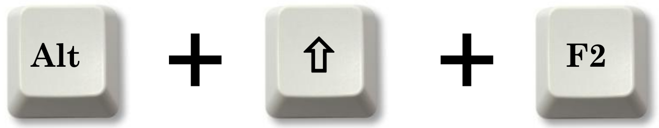
The contributing plug-in:

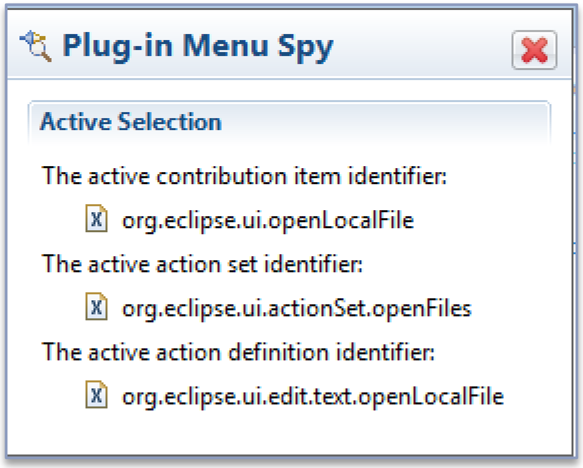
org.eclipse.jdt.ui (3.9.100.v20140219-1221)

The active view identifier:

org.eclipse.jdt.ui.PackageExplorer

Plug-in Menu Spy





Plug-in Menu Spy

Active Selection

The active contribution item identifier:

org.eclipse.ui.openLocalFile

The active action set identifier:

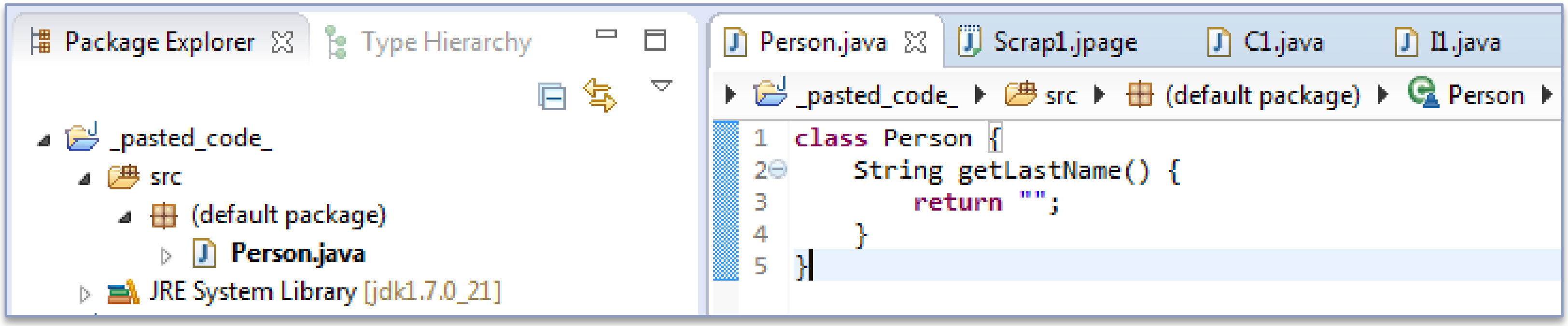
org.eclipse.ui.actionSet.openFiles

The active action definition identifier:

org.eclipse.ui.edit.text.openLocalFile

Paste code on Package Explorer

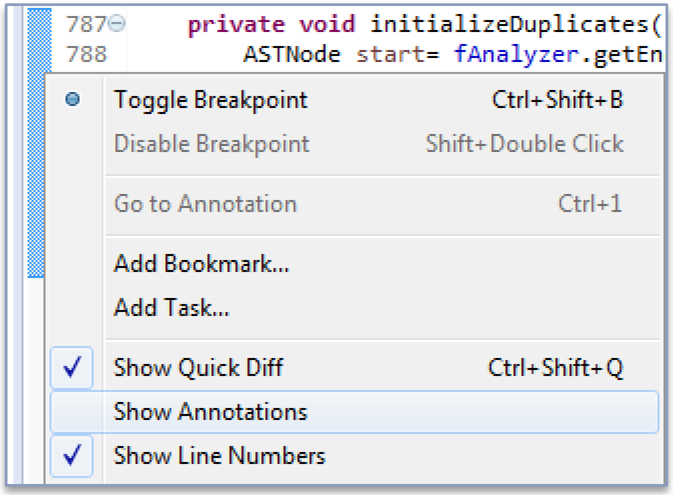
- Just copy the Java code and paste on Package Explorer.
- A new Java project will be created and the file will be opened in the Java editor.



```java
class Person {
    String getLastName() {
        return "";
    }
}
```

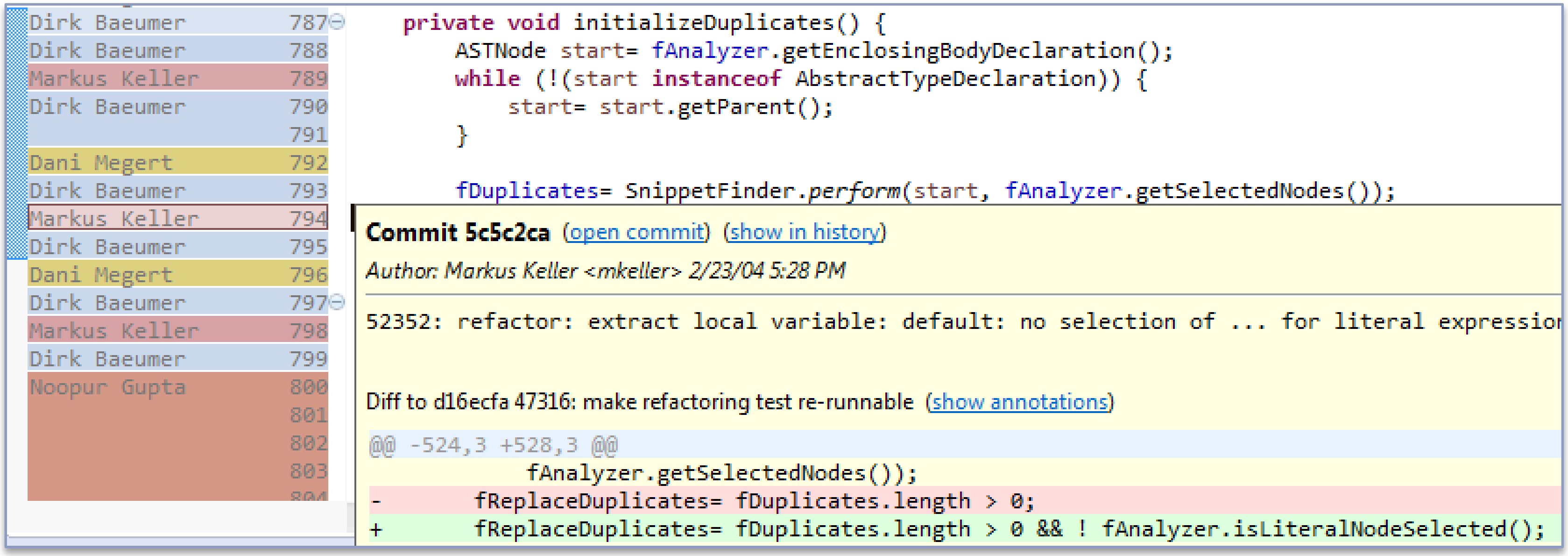
Show Annotations

- To determine who last modified a line of code and when.



```
787    private void initializeDuplicates(
788        ASTNode start= fAnalyzer.getEn
```

●	Toggle Breakpoint	Ctrl+Shift+B
	Disable Breakpoint	Shift+Double Click
	Go to Annotation	Ctrl+1
	Add Bookmark...	
	Add Task...	
✓	Show Quick Diff	Ctrl+Shift+Q
	Show Annotations	
✓	Show Line Numbers	



```
Dirk Baeumer    787    private void initializeDuplicates() {
Dirk Baeumer    788        ASTNode start= fAnalyzer.getEnclosingBodyDeclaration();
Markus Keller   789        while (!(start instanceof AbstractTypeDeclaration)) {
Dirk Baeumer    790            start= start.getParent();
                791        }
Dani Megert     792
Dirk Baeumer    793        fDuplicates= SnippetFinder.perform(start, fAnalyzer.getSelectedNodes());
Markus Keller   794
Dirk Baeumer    795
Dani Megert     796
Dirk Baeumer    797
Markus Keller   798
Dirk Baeumer    799
Noopur Gupta    800
                801
                802
                803
```

Commit 5c5c2ca (open commit) (show in history)

Author: Markus Keller <mkeller> 2/23/04 5:28 PM

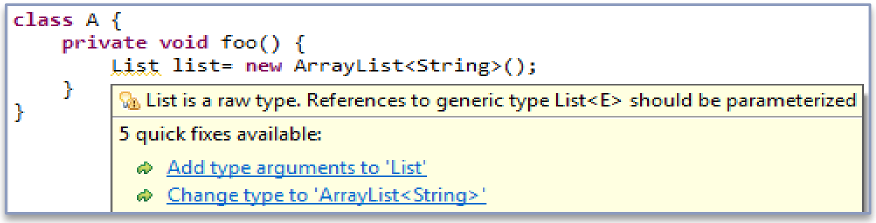
```
52352: refactor: extract local variable: default: no selection of ... for literal expressio
```

Diff to d16ecfa 47316: make refactoring test re-runnable (show annotations)

```
@@ -524,3 +528,3 @@
            fAnalyzer.getSelectedNodes());
-       fReplaceDuplicates= fDuplicates.length > 0;
+       fReplaceDuplicates= fDuplicates.length > 0 && ! fAnalyzer.isLiteralNodeSelected();
```

Hover

- When there is an error/warning at an identifier, the corresponding message is shown on hover instead of the Javadoc.



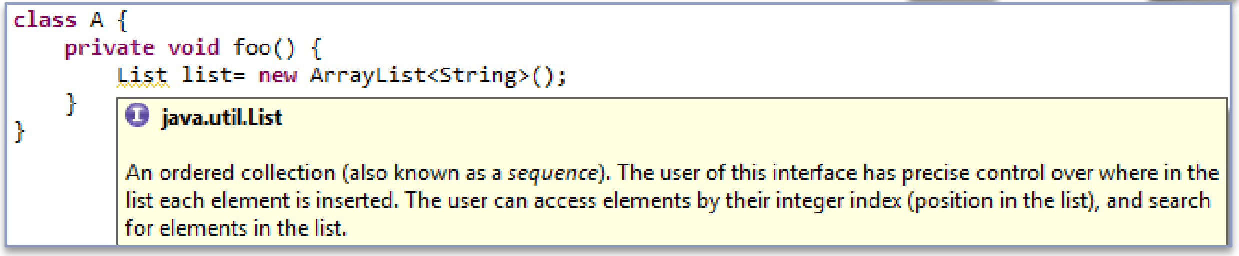
```
class A {
    private void foo() {
        List list= new ArrayList<String>();
    }
}
```

List is a raw type. References to generic type List<E> should be parameterized

5 quick fixes available:

- Add type arguments to 'List'
- Change type to 'ArrayList<String>'

- To see the Javadoc in such cases, press Ctrl + ⇧ and hover.



```
class A {
    private void foo() {
        List list= new ArrayList<String>();
    }
}
```

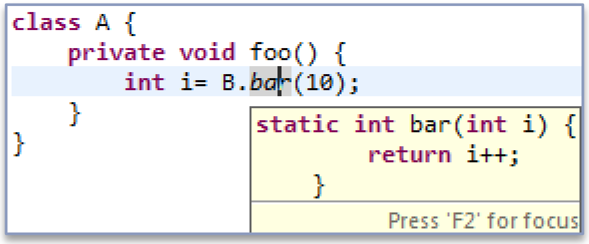
java.util.List

An ordered collection (also known as a *sequence*). The user of this interface has precise control over where in the list each element is inserted. The user can access elements by their integer index (position in the list), and search for elements in the list.

Or use Javadoc view.

- Update text hover key modifers at: ***Window > Preferences > Java > Editor > Hovers***

- To see the source on hover, press ⇧ and hover.



```
class A {
    private void foo() {
        int i= B.bar(10);
    }
}
```

```
static int bar(int i) {
        return i++;
    }
```

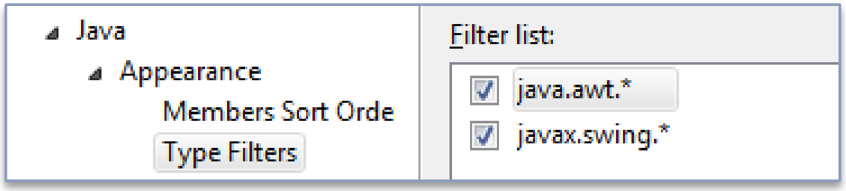
Press 'F2' for focus

Or use Declaration view.

Type Filters

- To ignore certain types from the Open Type dialog, content assist, quick fix proposals, import organization etc.

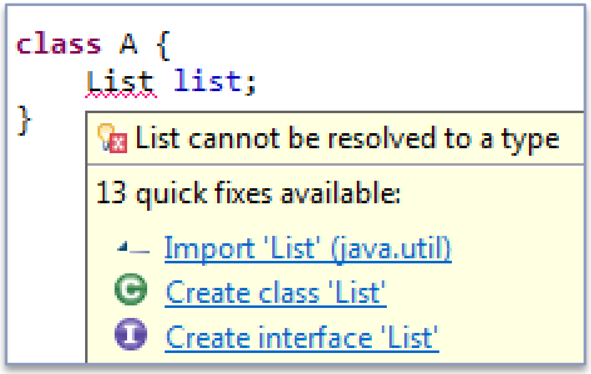
Window > Preferences > Java > Appearance > Type Filters


Java
Appearance
Members Sort Orde
Type Filters

Filter list:

✓ java.awt.*
✓ javax.swing.*

- Example:


```
class A {
    List list;
}
```

List cannot be resolved to a type

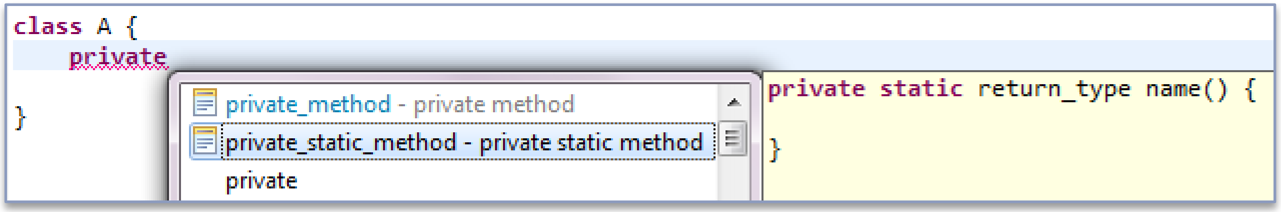
13 quick fixes available:

Import 'List' (java.util)
Create class 'List'
Create interface 'List'

Templates

- Templates are shown together with the Content Assist (Ctrl+Space) proposals.
- There are existing templates that you can configure or define new templates.

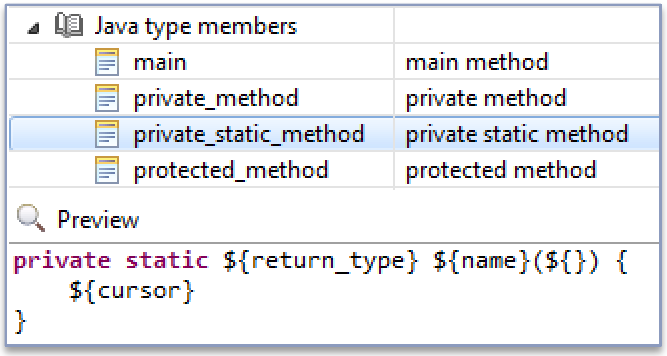
Window > Preferences > Java > Editor > Templates OR ***Templates view***



```
class A {
    private
}
```

private_method - private method
private_static_method - private static method
private

```
private static return_type name() {

}
```



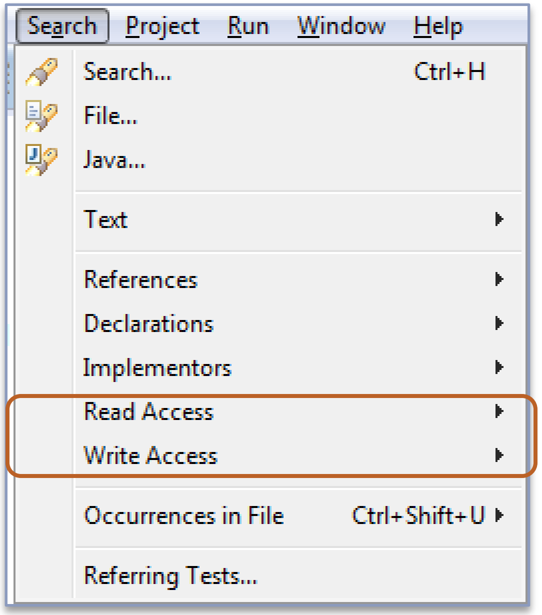
Java type members	
main	main method
private_method	private method
private_static_method	private static method
protected_method	protected method

Preview

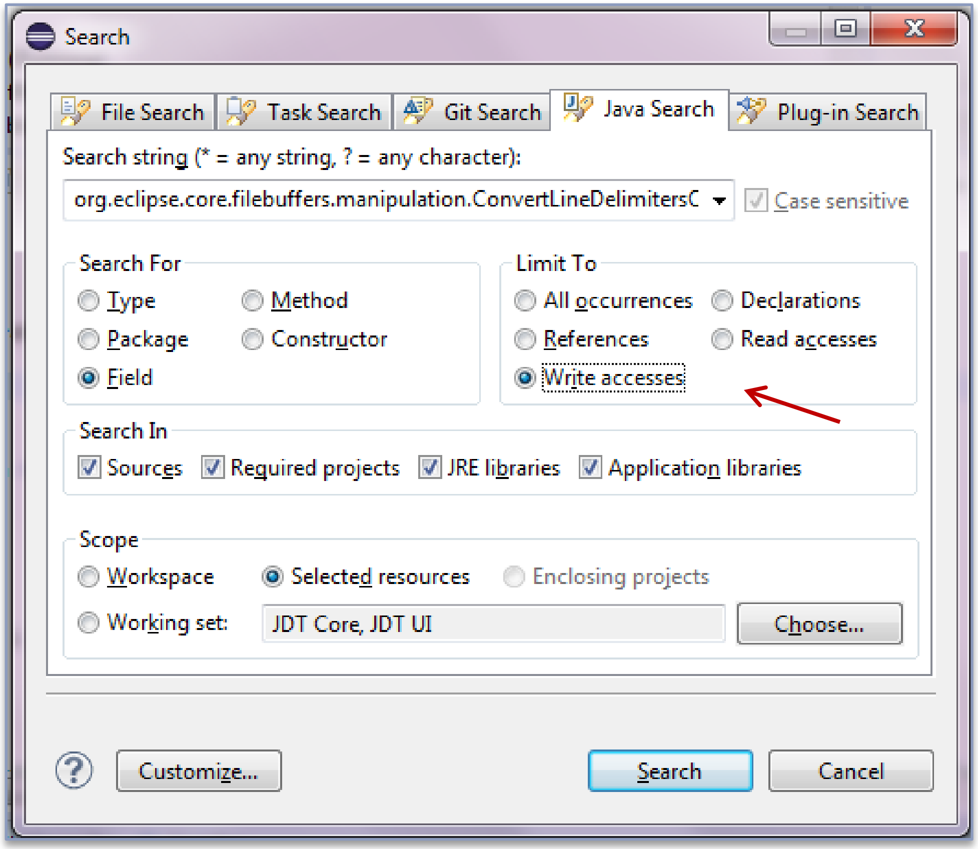
```
private static ${return_type} ${name}(${}) {
    ${cursor}
}
```

Search menu actions

- To find all read/write accesses to the selected field or local variable in the chosen scope.



OR



Search string (* = any string, ? = any character):
org.eclipse.core.filebuffers.manipulation.ConvertLineDelimitersC
[✓ Case sensitive]

Search For: Type, Method, Package, Constructor, [✓ Field]

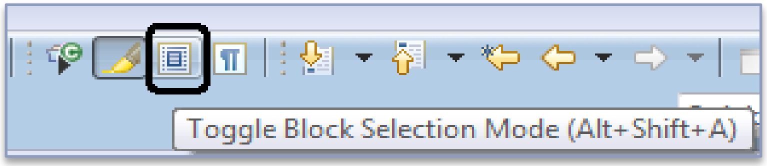
Limit To: All occurrences, Declarations, References, Read accesses, [✓ Write accesses]

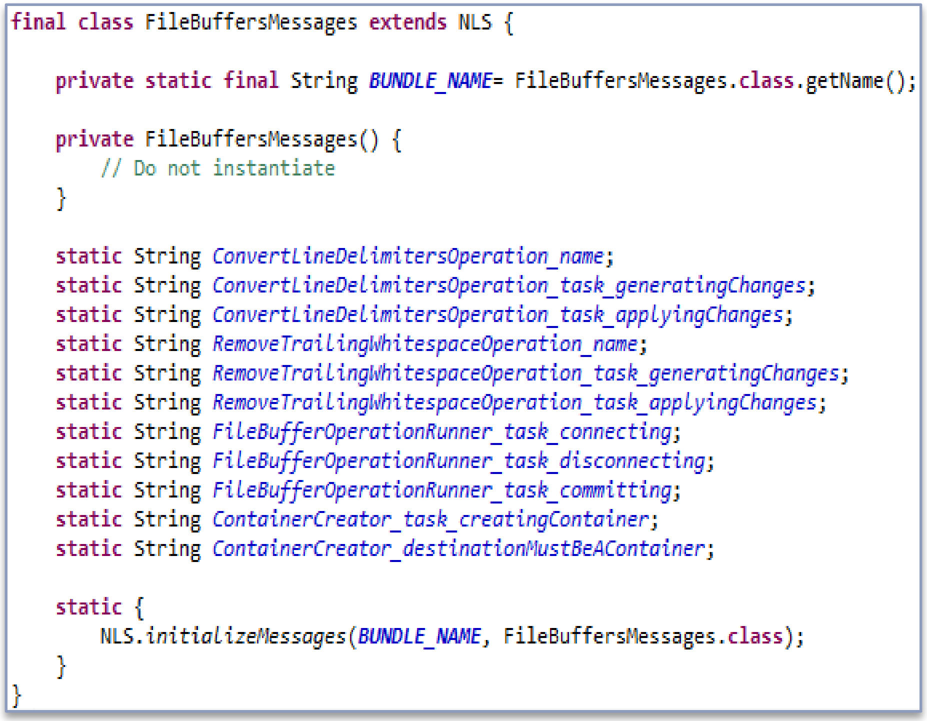
Search In: [✓ Sources] [✓ Required projects] [✓ JRE libraries] [✓ Application libraries]

Scope: Workspace, [✓ Selected resources], Enclosing projects, Working set: JDT Core, JDT UI

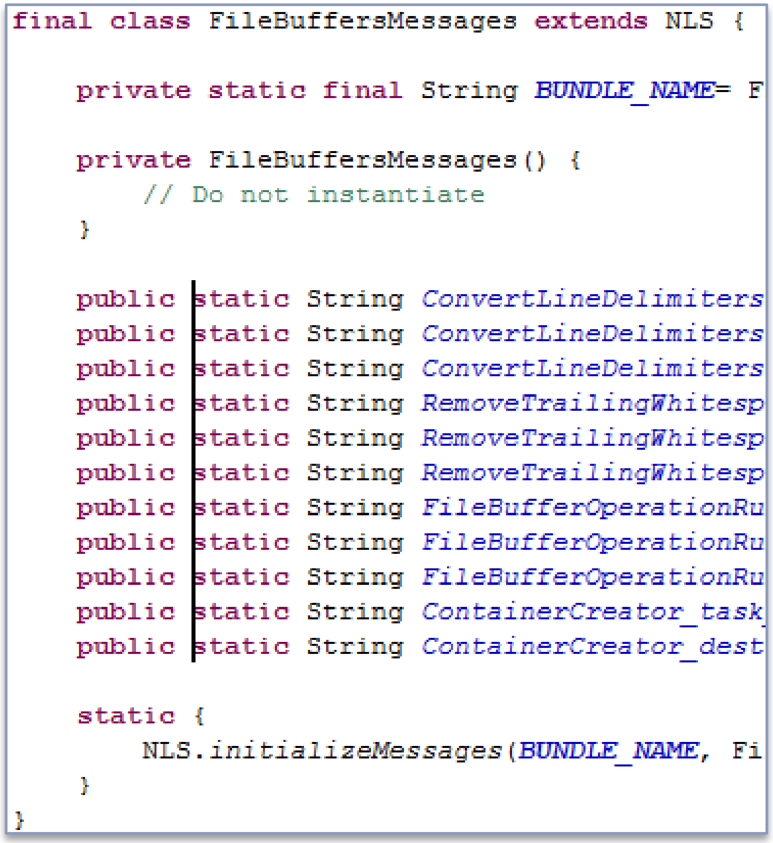
Block Selection

- To edit large number of almost identical lines.





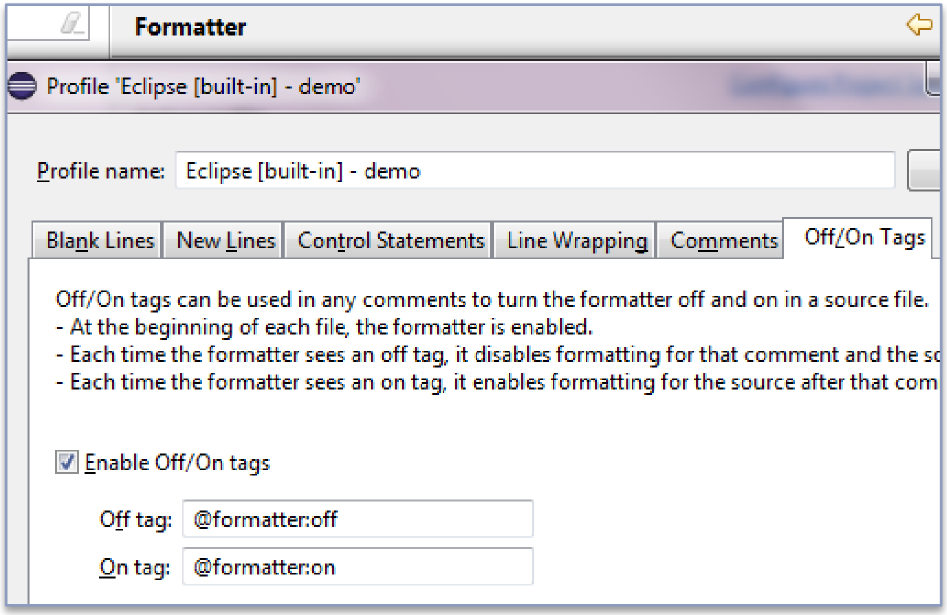
```
final class FileBuffersMessages extends NLS {

    private static final String BUNDLE_NAME= FileBuffersMessages.class.getName();

    private FileBuffersMessages() {
        // Do not instantiate
    }

    static String ConvertLineDelimitersOperation_name;
    static String ConvertLineDelimitersOperation_task_generatingChanges;
    static String ConvertLineDelimitersOperation_task_applyingChanges;
    static String RemoveTrailingWhitespaceOperation_name;
    static String RemoveTrailingWhitespaceOperation_task_generatingChanges;
    static String RemoveTrailingWhitespaceOperation_task_applyingChanges;
    static String FileBufferOperationRunner_task_connecting;
    static String FileBufferOperationRunner_task_disconnecting;
    static String FileBufferOperationRunner_task_committing;
    static String ContainerCreator_task_creatingContainer;
    static String ContainerCreator_destinationMustBeAContainer;

    static {
        NLS.initializeMessages(BUNDLE_NAME, FileBuffersMessages.class);
    }
}
```



```
final class FileBuffersMessages extends NLS {

    private static final String BUNDLE_NAME= F

    private FileBuffersMessages() {
        // Do not instantiate
    }

    public static String ConvertLineDelimiters
    public static String ConvertLineDelimiters
    public static String ConvertLineDelimiters
    public static String RemoveTrailingWhitesp
    public static String RemoveTrailingWhitesp
    public static String RemoveTrailingWhitesp
    public static String FileBufferOperationRu
    public static String FileBufferOperationRu
    public static String FileBufferOperationRu
    public static String ContainerCreator_task
    public static String ContainerCreator_dest

    static {
        NLS.initializeMessages(BUNDLE_NAME, Fi
    }
}
```

Formatter Off/On Tags

- Formatter Off/On tags can be used in any comment to turn the formatter off and on.
- Example: To prevent formatting of SQL queries.



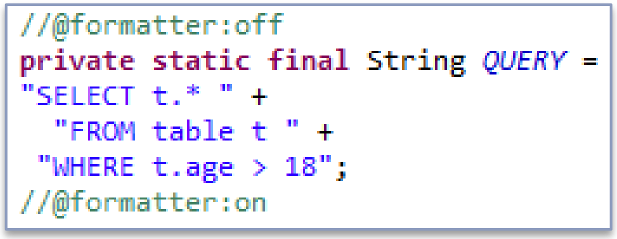
Profile name: Eclipse [built-in] - demo

Off/On tags can be used in any comments to turn the formatter off and on in a source file.
- At the beginning of each file, the formatter is enabled.
- Each time the formatter sees an off tag, it disables formatting for that comment and the so
- Each time the formatter sees an on tag, it enables formatting for the source after that com

☑ Enable Off/On tags

Off tag: @formatter:off

On tag: @formatter:on



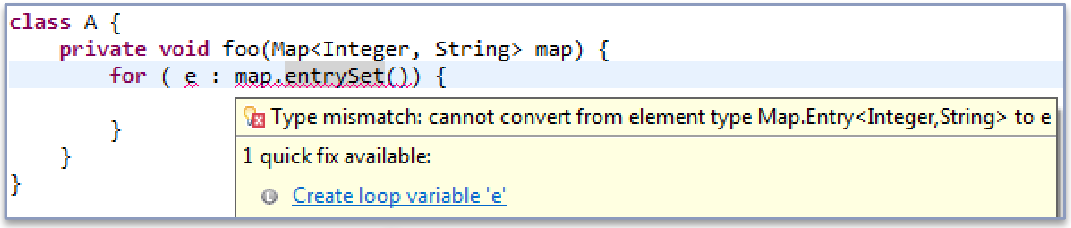
```
//@formatter:off
private static final String QUERY =
"SELECT t.* " +
  "FROM table t " +
 "WHERE t.age > 18";
//@formatter:on
```

Quick Fixes and Quick Assists

Ctrl + 1

- Examples:

Quick Fix:

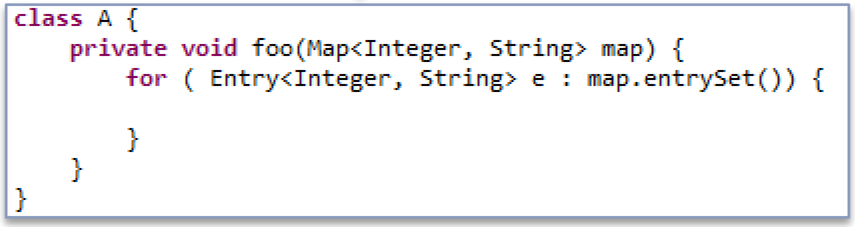


```
class A {
    private void foo(Map<Integer, String> map) {
        for ( e : map.entrySet()) {

        }
    }
}
```

Type mismatch: cannot convert from element type Map.Entry<Integer,String> to e

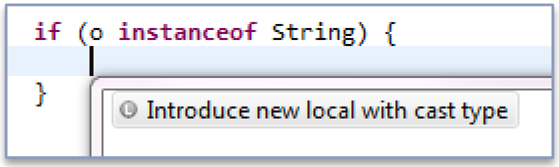
1 quick fix available:

Create loop variable 'e'



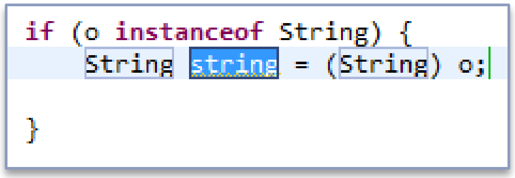
```
class A {
    private void foo(Map<Integer, String> map) {
        for ( Entry<Integer, String> e : map.entrySet()) {

        }
    }
}
```

Quick Assist:



```
if (o instanceof String) {

}
```

Introduce new local with cast type



```
if (o instanceof String) {
    String string = (String) o;

}
```

- Don’t type too much yourself – Let Eclipse help you with quick fixes, quick assists, refactorings, content assist and more.

Keyboard Shortcuts

Ctrl + ⇧ + L = Lists all keyboard short cuts

- Examples:

Ctrl + M = Maximize/Minimize Editor/View

Ctrl + ⇧ + X / Y = To upper case/ lower case

Alt + ↑ / ↓ = Move line(s)

Ctrl + D = Delete current line

Ctrl + / = Comment/Uncomment line

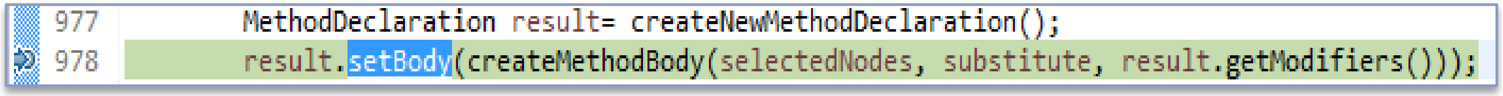
Ctrl + E = Quick Switch Editor

Alt + ⇧ + ↑ = Expand selection to enclosing element

- **Debugging**
  - Smart Step Into Selection
  - Types of Breakpoints
  - Step Filters
  - Scrapbook Page

Smart Step Into Selection

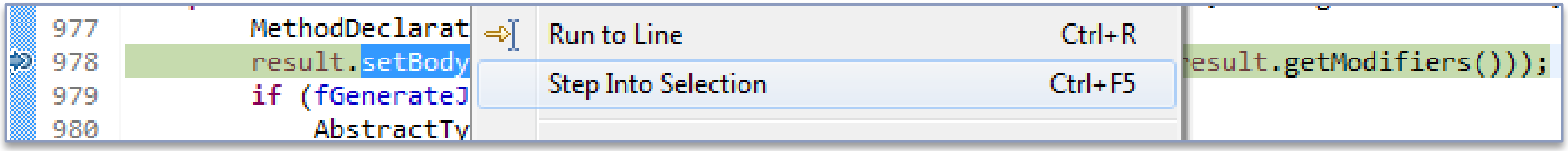
- To step into a single method within a series of chained or nested method calls.
- Example:



```
        MethodDeclaration result= createNewMethodDeclaration();
        result.setBody(createMethodBody(selectedNodes, substitute, result.getModifiers()));
```

Ctrl + F5 OR Ctrl + Alt + Click

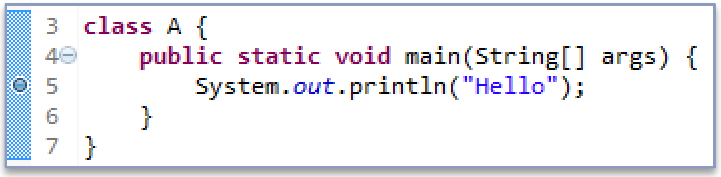
OR



```
        MethodDeclarat
        result.setBody                                    result.getModifiers()));
        if (fGenerateJ
            AbstractTy
```

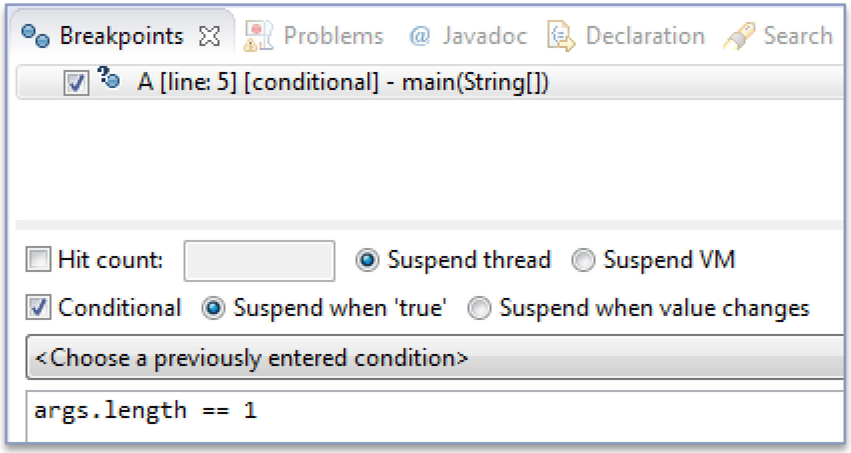
Types of Breakpoints

- Line Breakpoint



```
class A {
    public static void main(String[] args) {
        System.out.println("Hello");
    }
}
```

- Conditional Breakpoint



☑ A [line: 5] [conditional] - main(String[])

☐ Hit count: [● Suspend thread] ○ Suspend VM

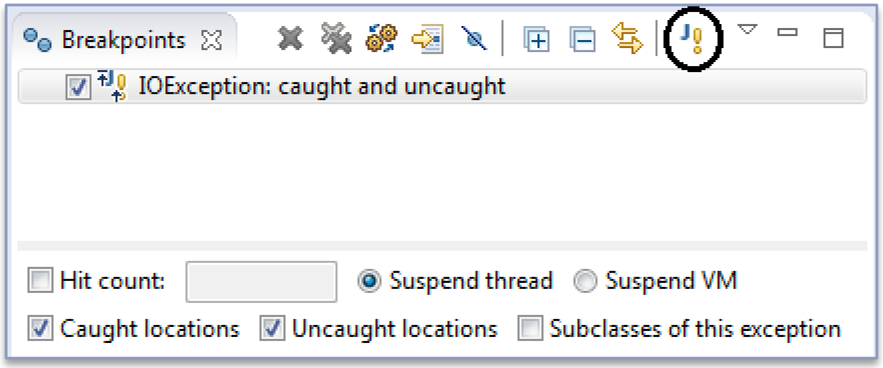
☑ Conditional [● Suspend when 'true'] ○ Suspend when value changes

<Choose a previously entered condition>

```
args.length == 1
```

- Exception Breakpoint:

  When exceptions are passed over several layers, they are often wrapped or discarded in another exception. To find the origins of an exception, use Exception breakpoint. The execution will suspend whenever the exception is thrown or caught.



☑ IOException: caught and uncaught

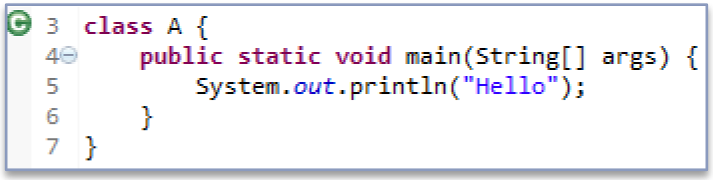
☐ Hit count: [● Suspend thread] ○ Suspend VM

☑ Caught locations ☑ Uncaught locations ☐ Subclasses of this exception

Types of Breakpoints

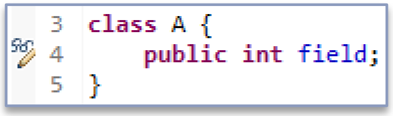
- Classload Breakpoint:

  To inspect who is trying to load the class or where is it used for the first time.



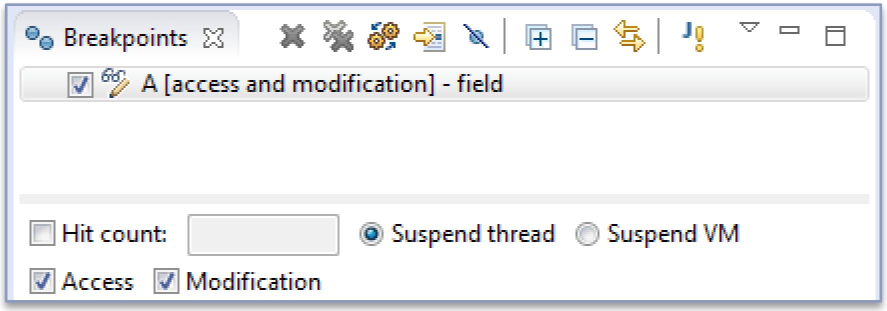
```
class A {
    public static void main(String[] args) {
        System.out.println("Hello");
    }
}
```

- Watchpoint:



```
class A {
    public int field;
}
```

  To suspend the execution where a field is accessed or modified.



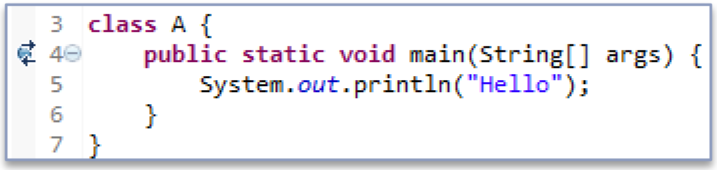
Breakpoints

[✓] A [access and modification] - field

[ ] Hit count: [✓ Suspend thread] Suspend VM

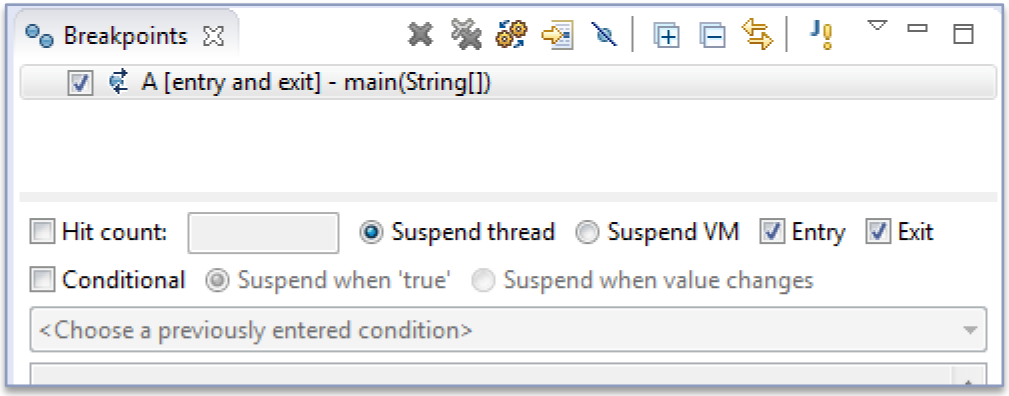
[✓] Access [✓] Modification

- Method Breakpoint:



```
class A {
    public static void main(String[] args) {
        System.out.println("Hello");
    }
}
```

  To suspend the execution when the method is entered or exited.



Breakpoints

[✓] A [entry and exit] - main(String[])

[ ] Hit count: [✓ Suspend thread] Suspend VM [✓] Entry [✓] Exit

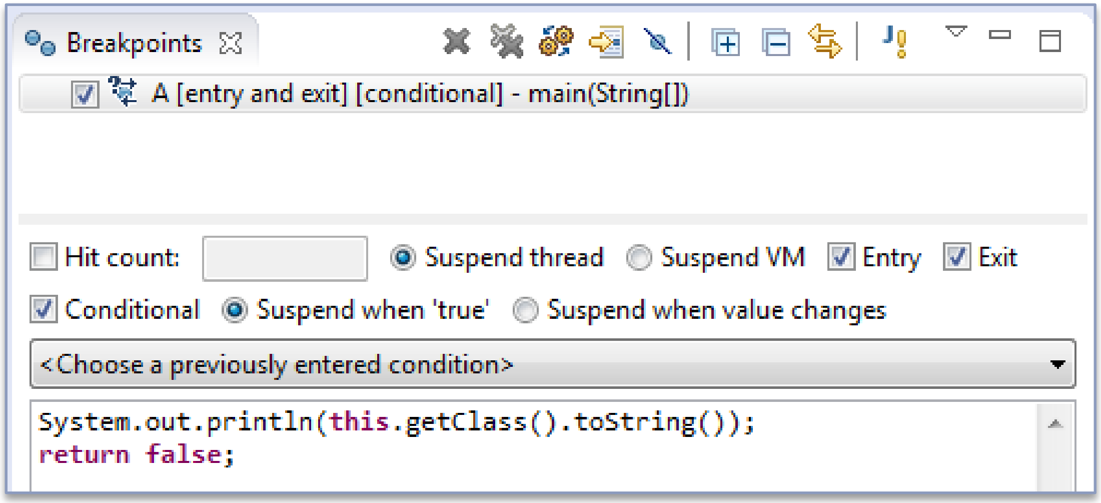
[ ] Conditional [✓ Suspend when 'true'] Suspend when value changes

<Choose a previously entered condition>

Printpoint: A *point* in code where the debugger does not *break* the execution but only *prints* to console.

- To debug race conditions or to see the order of threads execution.
- To prevent the addition of print statements in the code while debugging.

Set a conditional breakpoint with ***Suspend when 'true'*** option and a condition which is always false (eg: ***return false;***) as the last statement.



A [entry and exit] [conditional] - main(String[])

[ ] Hit count: [● Suspend thread] ○ Suspend VM [✓ Entry] [✓ Exit]

[✓ Conditional] [● Suspend when 'true'] ○ Suspend when value changes

<Choose a previously entered condition>

```
System.out.println(this.getClass().toString());
return false;
```

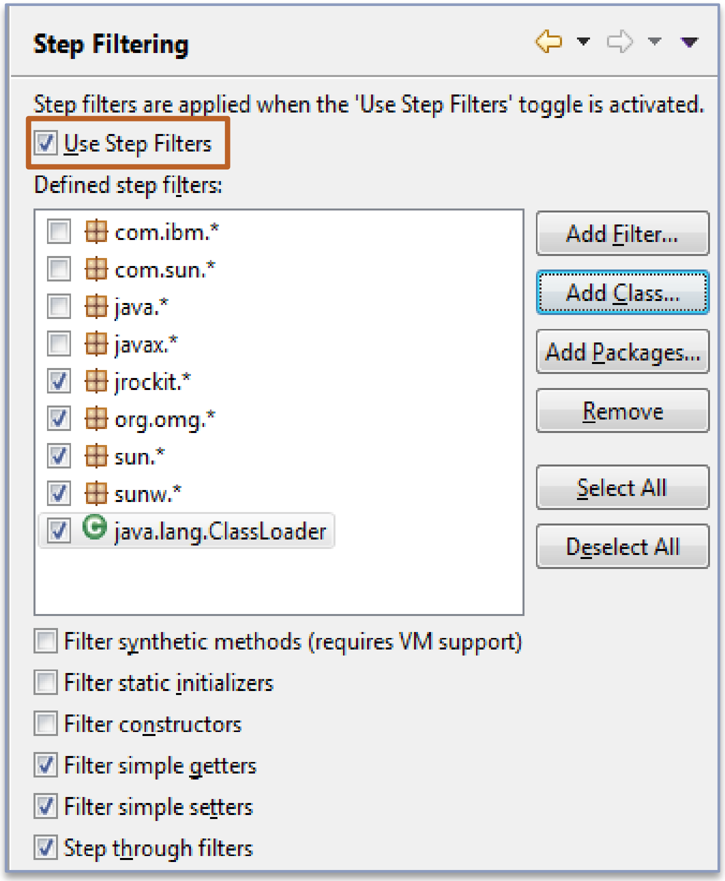
Step Filters

- To filter out specified classes and packages while steping into code during debugging.

Window > Preferences > Java > Debug > Step Filtering

OR

In the Debug view, the selected stack frame's package or declaring type can be quickly added to the list of filters by selecting ***Filter Type*** or ***Filter Package*** from the stack frame's context menu.



Step Filtering

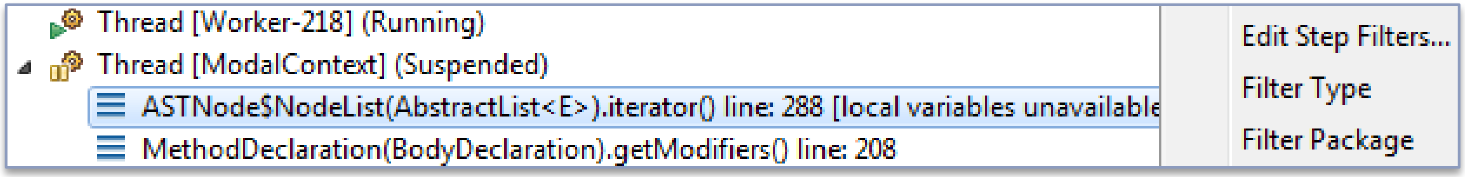
Step filters are applied when the 'Use Step Filters' toggle is activated.

- [✓] Use Step Filters

Defined step filters:

- [ ] com.ibm.*
- [ ] com.sun.*
- [ ] java.*
- [ ] javax.*
- [✓] jrockit.*
- [✓] org.omg.*
- [✓] sun.*
- [✓] sunw.*
- [✓] java.lang.ClassLoader

- [ ] Filter synthetic methods (requires VM support)
- [ ] Filter static initializers
- [ ] Filter constructors
- [✓] Filter simple getters
- [✓] Filter simple setters
- [✓] Step through filters

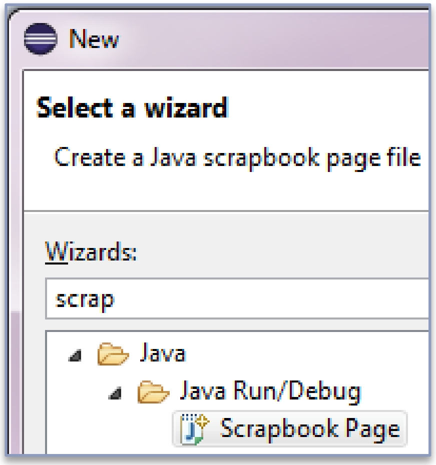


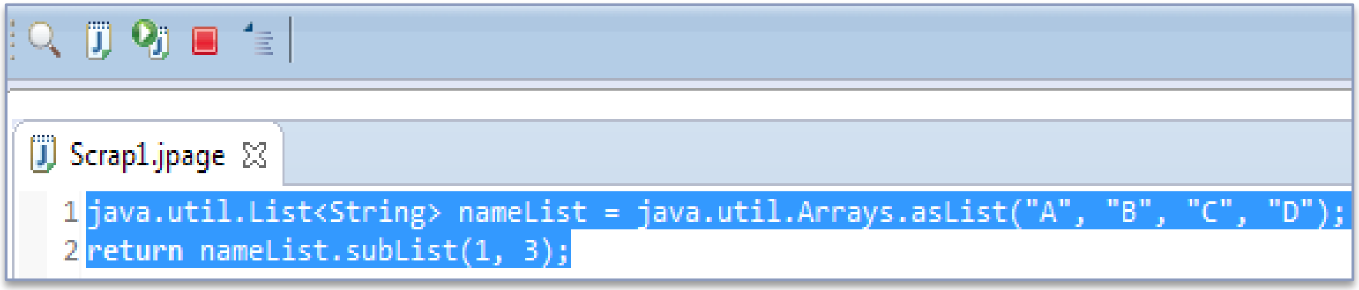
```
Thread [Worker-218] (Running)
Thread [ModalContext] (Suspended)
    ASTNode$NodeList(AbstractList<E>).iterator() line: 288 [local variables unavailable
    MethodDeclaration(BodyDeclaration).getModifiers() line: 208
```

Scrapbook Page: A container for random snippets of code that can be executed any time without a context.

- To experiment with an API or test a piece of code (algorithm/method).

- No need to create a new project / class / main method / run the application to test.



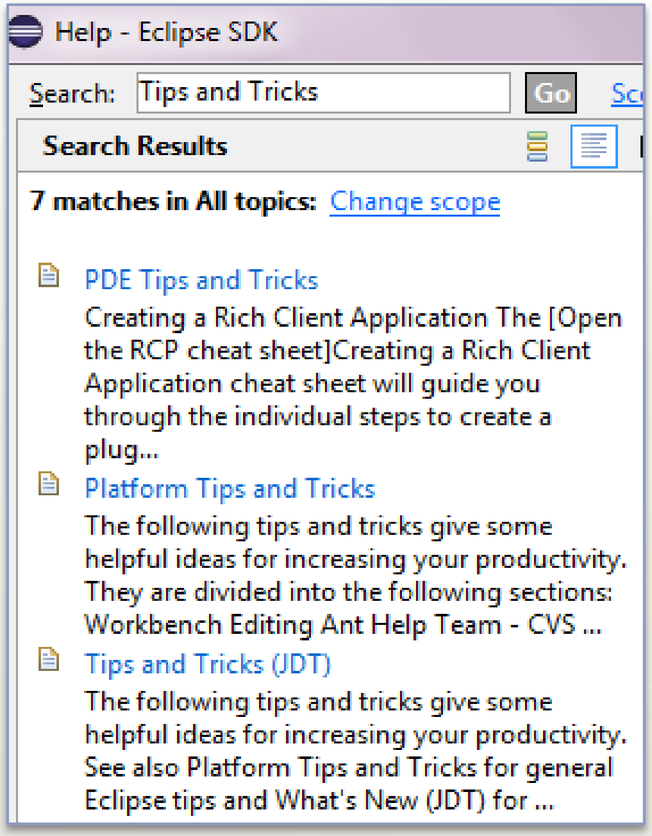


```
java.util.List<String> nameList = java.util.Arrays.asList("A", "B", "C", "D");
return nameList.subList(1, 3);
```

LAST BUT NOT THE LEAST!

Help > Tips and Tricks...

Help > Help Contents > Search "*What's new*"



Help - Eclipse SDK

Search: Tips and Tricks Go

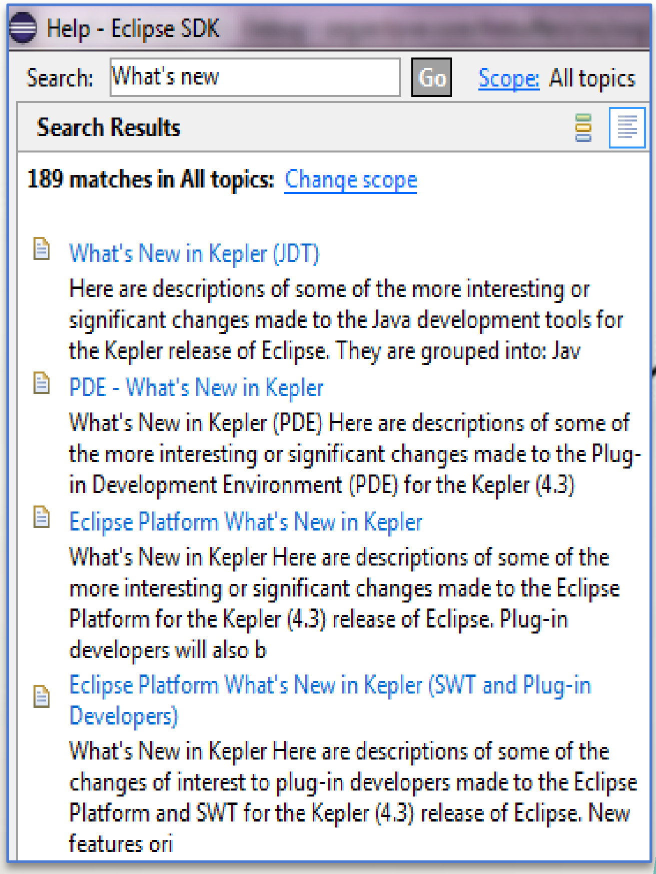
Search Results

7 matches in All topics: Change scope

PDE Tips and Tricks
Creating a Rich Client Application The [Open the RCP cheat sheet]Creating a Rich Client Application cheat sheet will guide you through the individual steps to create a plug...

Platform Tips and Tricks
The following tips and tricks give some helpful ideas for increasing your productivity. They are divided into the following sections: Workbench Editing Ant Help Team - CVS ...

Tips and Tricks (JDT)
The following tips and tricks give some helpful ideas for increasing your productivity. See also Platform Tips and Tricks for general Eclipse tips and What's New (JDT) for ...



Help - Eclipse SDK

Search: What's new Go Scope: All topics

Search Results

189 matches in All topics: Change scope

What's New in Kepler (JDT)
Here are descriptions of some of the more interesting or significant changes made to the Java development tools for the Kepler release of Eclipse. They are grouped into: Jav

PDE - What's New in Kepler
What's New in Kepler (PDE) Here are descriptions of some of the more interesting or significant changes made to the Plug-in Development Environment (PDE) for the Kepler (4.3)

Eclipse Platform What's New in Kepler
What's New in Kepler Here are descriptions of some of the more interesting or significant changes made to the Eclipse Platform for the Kepler (4.3) release of Eclipse. Plug-in developers will also b

Eclipse Platform What's New in Kepler (SWT and Plug-in Developers)
What's New in Kepler Here are descriptions of some of the changes of interest to plug-in developers made to the Eclipse Platform and SWT for the Kepler (4.3) release of Eclipse. New features ori

Thank You

Evaluate This Session

1 Sign-in: www.eclipsecon.org

2 Select session from schedule

3 Evaluate: +1 0 -1