1. Installing plugin with Eclipse's update manager

If you are using Eclipse 3.0 or higher you can use the update manager to automatically download and install the QJ-Pro plugin.

- Start Eclipse, go to the Help menu->Software Updates->Find and install. Then add a new remote update site and use http://qjpro.sourceforge.net/update_site as url location.
- Obey the dialog prompts, and restart Eclipse. That's all there is to it!

2. Installing plugin for Eclipse/Websphere(download)

If you are an Eclipse version < 3.0 or you want to manually download the plugin you need to unzip the zip file called <code>qjpro_Vx.x_eclipse.zip</code> in the plugins directory of your Eclipse directory. After you restart Eclipse you are ready to use QJ-Pro in the Java perspective.

3. Launching QJ-Pro from Eclipse

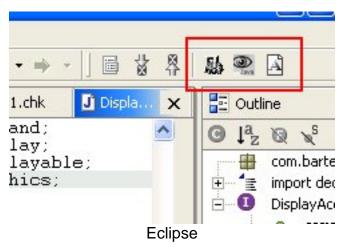
Through the Integration package, QJ-Pro becomes an integral part of the Eclipse IDE. To see the QJ-Pro Obervations you need to activate the QJ-Pro View:

Window => Show View => Other ... => QJ-Pro=>QJ-Pro. The functionality will be automatically available within the IDE at installation of the Eclipse Integration for QJ-Pro.

The QJ-Pro functionality can be accessed through the QJ-Pro-buttons or using the mouse.

QJ-Pro-buttons

The QJ-Pro-buttons are only active when the editor shows a java source file and the editor has the focus.



The first button is used to inspect the selected Java source file. The second button is used to inspect the complete Java project. The third button is used to show the annotated source file of the selected Java source file.

Mouse usage

If you press the right mouse button on a Java source file, Java package or Java project, a pop-up menu will appear with the standard Eclipse options extended with a QStudio menu. This QStudio menu includes the following menu options depending on the context: Inspect An analysis run is started for the selected file or node in the browser i.e. project, package or Java file. Annotated View A browser is launched showing the selected Java source file annotated with the observations, resulting from the last analysis run.

le Edit Source F	Refactor Navigate Search Pi	roject D	un Windo	w Help					
	an Lawrence and the Lawrence Lawrence				Li se li re	1			
Š • 🛛 🖳 🖞	} ☆・★・ @ 6	1 争 征	D Q -	A .	•] 🌤] 🗇 •	• ⇒ • 🖬 👔 .	¥ \$\$ 🖬 🎽		
Package Exp	olorer 🔍 🗙 🗋 Welco	ome	၂၂) Comma	nd 🗌 Comr	nand 🗋 Test1	.chk 🚺 Displa	X 🗄 Outline :		
(+ + E)					lcdui.Comma		O la R S		
J MicroEmulator 0.					lcdui.Displ lcdui.Displ		com.barteo		
the com.barteo.c					lcdui Graph				
com.barteo							DisplayAcce		
🖶 com.barteo	New) in	terfac	or comme					
👧 com.barteo	Open		: interface DisplayAccess						
🛨 🞣 Commai	Open With			🔷 🔴 keyPre					
🕂 🗹 🚺 Display/	Open Type Hierarchy	, 1q (id commandAction(Command cmd); <pre></pre>						
🗄 🗓 Display	open type hierarchy	spla							
Emulate	Cut	11/1		🛛 🔵 getCu					
	Сору	14	keyfres	sed(int ke	eylode);		 setCu updation 		
	Paste	id I	e upuau						
HIER NUMBER			id paint(Graphics q);						
⊕	Delete		+/C	manhina al					
	Delete Refactor	, id ;	paint(G	raphics g)	3	~	8		
🛨 🚺 MIDletE		, id ;		raphics g)	;				
	Refactor				;	<u>></u>			
MIDletE Ommai Commai Ommai Ommai Commai Commai Ommai Commai	Refactor Import Export				:	>	×		
MIDlete Commai Commai Commai Commai Commai Display	Refactor	for Ja		Access.java	:				
MIDletE Comman Comman Comman Comman Comman Display Display	Refactor Import Export	for Ja	ava - Display	Access.java	; Impact	Observation			
MIDletE Commai Commai Commai Display Display Display	Refactor Import Export Organize Imports	for Ja	ava - Display	Access.java		Observation	\$∎¶? - :		
MIDlete Commar Commar Commar Commar Commar Commar Display/	Refactor Import Export Organize Imports References Declarations	for Ja s In ine t0 :8	ava - Display Inspection Lo Pos 1 1	Access.java	Impact 2 3	Observation This file contains to Public declaration "	→ DisplayAccess" has no jay		
MIDlete Commar Commar Commar Commar Commar Display Display Display Display Display	Refactor Import Export Organize Imports References	for Ja 5 In 10 13 11	ava - Display Inspection Lo Pos 1 1 1	Access.java	Impact 2 3 1	Observation This file contains to Public declaration " Do not use tabs, la	The second secon		
MIDlete Commar Commar Commar Commar Commar Displayr Displayr Displayr Displayr	Refactor Import Export Organize Imports References Declarations	for Ja s In ine t0 :8	ava - Display Inspection Lo Pos 1 1	Attr conc clar env env	Impact 2 3	Observation This file contains to Public declaration " Do not use tabs, la	The second secon		
MIDlete Commar Commar Commar Display/ Display	Refactor Import Export Organize Imports References Declarations Refresh	for Ja for Ja for Ja ine ine ine ina	ava - Display Inspection Lo Pos 1 1 1	Access.java	Impact 2 3 1	Observation This file contains to Public declaration " Do not use tabs, la	The second secon		
MIDlete Comman Comman Comman Comman Comman Display Display Display Display Display Display Display Display Display	Refactor Import Export Organize Imports References Declarations Refresh Team	for Ja s In ine ine ine ine ine ine ine ine ine in	ava - Display Inspection Lo Pos 1 1 1	Access.java g Attr conc clar env env	Impact 2 3 1	Observation This file contains to Public declaration " Do not use tabs, la	→ DisplayAccess" has no jay		
MIDlete Comman Comman Comman Comman Comman Display Display Display Display Display Display Display	Refactor Import Export Organize Imports References Declarations Refresh Team Compare With	for Ja for Ja s In b ine b 10 :8 i1 b i3 b c 4	ava - Display inspection Lo Pos 1 1 1 1 1	Access.java g Attr conc clar env env	Impact 2 3 1	Observation This file contains to Public declaration " Do not use tabs, la	The second secon		

Eclipse inspect

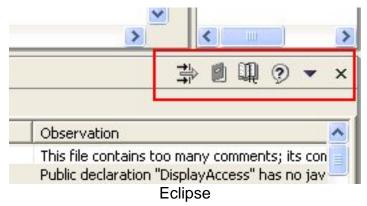
On the project node you can set the project properties and you can save the project. This menu options allows you to create a QJ-Pro project file which can be used for the Standalone version of QJ-Pro

Java - Eclipse Plat Edit Source Refa	ttorm ttor Navigate Search Proje	ct Run Windo	w Help					
š • 🛛 🖳 🗛]	爸!	• \$ \$ \$	A .] 🌤] 🗇	• ⇒ -] 🖻 🕈 🚇	\$ 2		
Package Explore Package Explore MicroEmulate Com.bai Com Disc Com Disc Com Disc Com Disc Com Disc Com Disc Disc	import import	i javax mic javax mic avax mic avax mic nterfac command ay getD: keyPres	Command] Command] Test1.chk] Displa × javax.microedition.lcdui.Command; javax.microedition.lcdui.Display;					
	Import Export Organize Imports	paint(G						
	Refresh Close Project							
	Team Compare With Replace With Restore From Local History	Pos 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	Attr conc clar env env	Impact 2 3 1 1	Public declaration "Disp Do not use tabs, layou	o many comments; its con isplayAccess" has no jav out may differ when view out may differ when view		
Package Explorer	QStudio Properties	Left Inspect				>		
		Save a:	s QStudio Project					

4. The Results Panes of QJ-Pro.

Eclipse's main window consists of 4 panes. At the bottom we find the QJ-Pro pane besides the Tasks, Log Console and Console pane. The QJ-Pro Integration uses the QJ-Pro pane to show the processing results. In QJ-Pro pane 2 tabs are used the "Observations"- and "Inspection Log" tabs. The Observation tab shows a table of observation messages and related information. The Inspection Log tab shows the warnings/errors which occur during an analysis.

At the top on the rightside of the QJ-Pro pane you will find 4 icons.



The first icon is a shortcut to the setting of "Selection View". By default, the Observation Pane shows all observations of an analysis run. For convenience a view can be selected to limit the number of observations shown. See the tutorial on the Native client for a more detailed description of this pane. The second icon is a shortcut to the Rules Guide. The third icon is a shortcut to the Pattern Guide. The fourth icon is a shortcut to the User Manual.

5. The Observation Pane

The results of an analysis run are presented in a table. Each row relates to an observation. The following information is provided:

- ID: The identification number of the rule. This is a unique number that identifies the rule, all rules are specified in the rules guide.
- Line: The line number within the Java source file where the observation originates.
- Pos: The character position within the Java source code line where the observation originates.
- Attr: The quality sub-attribute that is assigned to the rule. See the concepts part for more information on quality (sub)-attributes.
- Impact: The impact level that is assigned to the rule. See the concepts part for more information on impact levels.

The columns can be sorted by clicking on the label of the column and a double clicking reverses the ordering.

The Eclipse Source Editor is synchronized by selecting a row in the observation table. The Source Editor Pane will show the code lines where the observation refers to.

6. The Inspection Log Pane

This window will contain all the errors which occurred during an inspection run. Click the right mouse button in the dialog to clear the inspection log. See the tutorial on the Native

client for a more detailed description of this pane.

7. The Status bar

At the very bottom of the screen there is a status bar that shows the state of the tool.