

# Enabling the Unit Test Assistant Interface

UTA includes several views that allow you to create and enhance your tests, as well as review test results. You can open the views by choosing **Parasoft> Show View** and selecting the view you want to open. Alternatively, you can open the Jtest perspective, which includes all of the UTA views (see [Opening the Jtest Perspective in Eclipse](#)).

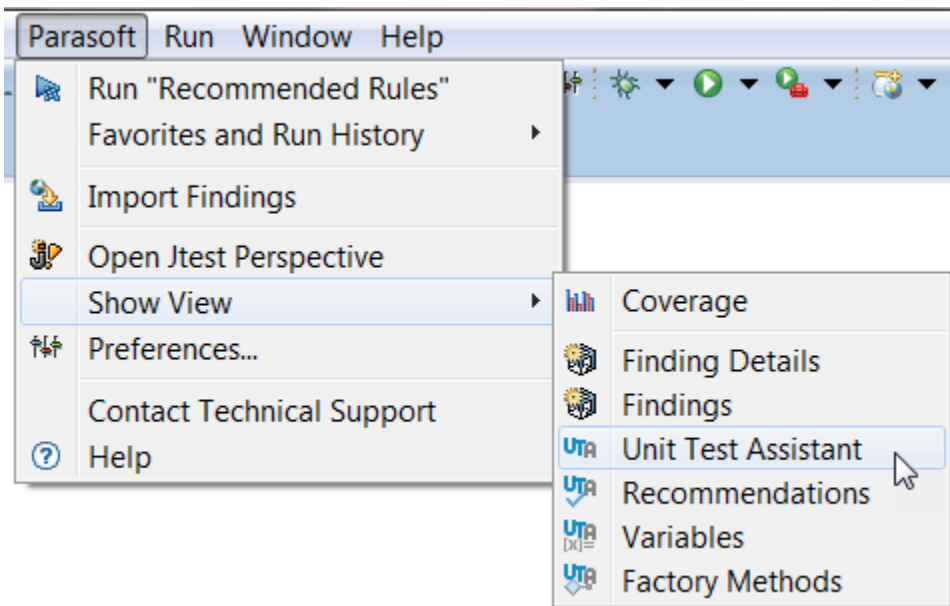
The following views are available:

- [Unit Test Assistant View](#)
- [Recommendations View](#)
- [Variables View](#)
- [Factory Methods View](#)

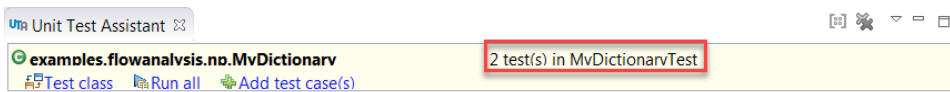
## Unit Test Assistant View


The Unit Test Assistant view is a dynamic interface that allows you to create, run, and monitor unit tests. It can detect tests that have been created for the class or method selected in the editor and display actions available for this particular class or method.

1. Choose **Parasoft> Show View> Unit Test Assistant** to enable the Unit Test Assistant interface.

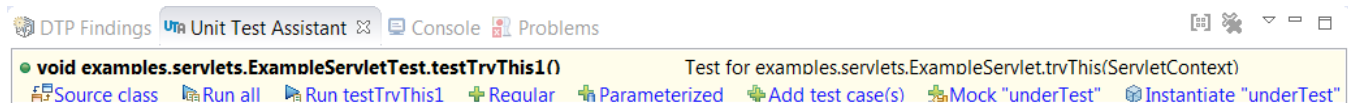


2. Open a source file in the editor and click a class or method to display available actions and the number of tests that have been created for it.

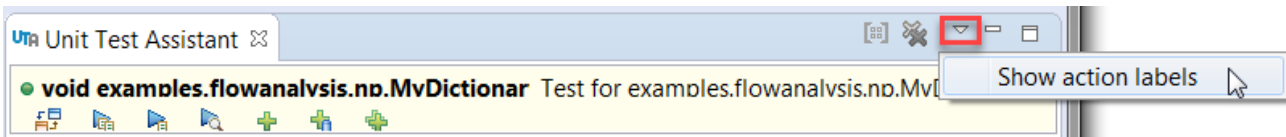


3. Switch between the test and the source class with the  **Source class** and  **Test class** buttons.

The Unit Test Assistant view is context-sensitive and displays actions that are available for the current selection in the editor (see [About UTA Actions](#) for details). You can perform an action by clicking on its label or icon.



You can enable or disable showing action labels by selecting or deselecting the **Show action labels** option in the UTA interface menu. If disabled, the Unit Test Assistant view only displays action icons:

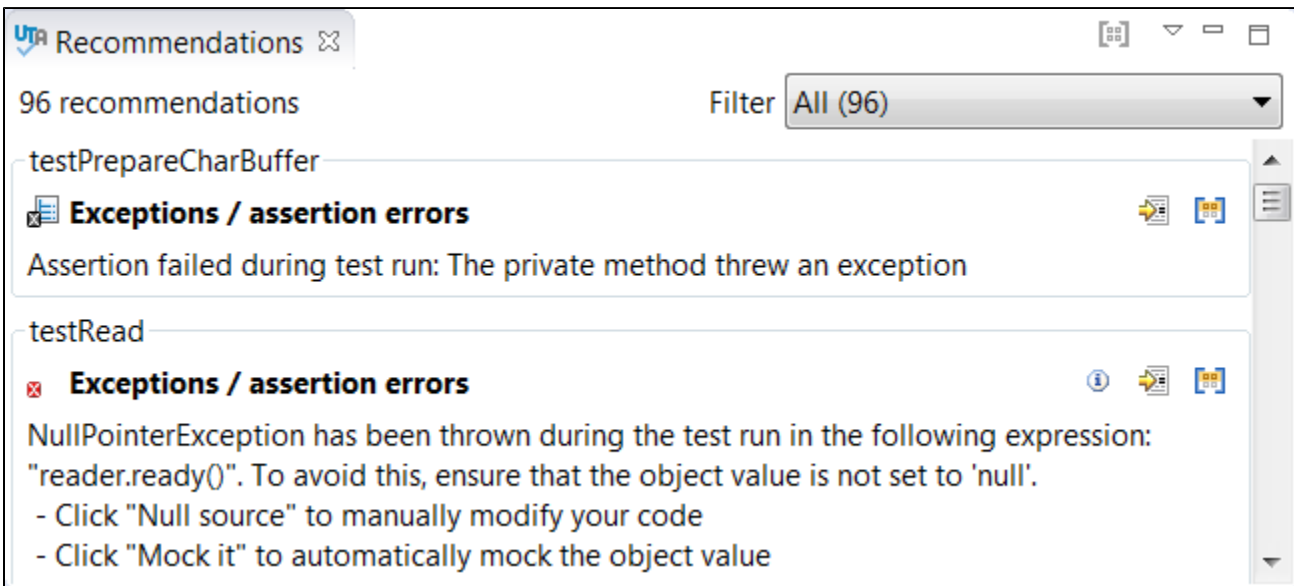


## Monitoring Execution Flow

The Unit Test Assistant View allows you to monitor test execution flow and conveniently navigate your code; see [Executing Unit Tests with Unit Test Assistant](#).

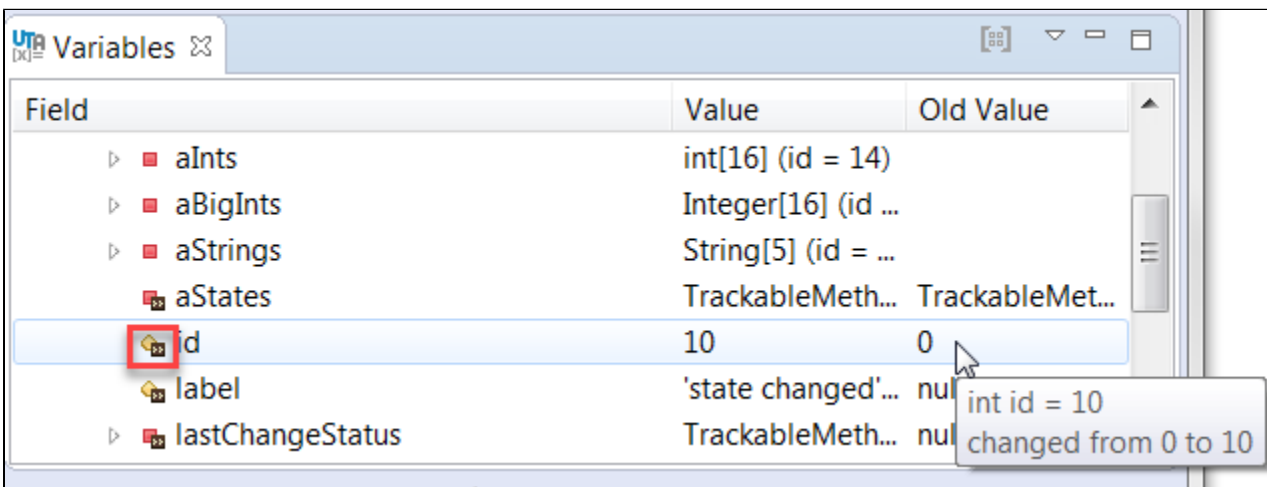
## Recommendations View

This view displays recommendations for test improvements after a test(s) is run; see [Working with Recommendations](#) for details.



## Variables View

This view displays variables associated with method calls. You can use this view to compare the state of an object before and after a method call. The >> icon indicates the variables whose values changed during execution. See [Tracking Object Changes and Creating Assertions](#) for details.



## Factory Methods View

This view provides an interface for scanning your code for methods that have been tagged as factory methods in the Javadoc (see [Configuring Factory Methods](#) for details) and display them in a tree.

