

WindowTester API Quick Reference

UIContext Methods

```
ILocator click(ILocator locator)
    click the widget identified by this locator
ILocator contextClick(ILocator locator, String path)
    select the given context menu path on this widget
void enterText(String text)
    enter the given text
IWidgetLocator find(ILocator locator)
    find the widget identified by this locator
IWidgetLocator[] findAll(ILocator locator)
    find all the widgets identified by this locator
IWidgetLocator dragTo(ILocator locator)
    drag from the current hover position to this locator
void wait(ICondition condition)
    wait for this condition to become true
void assertThat(ICondition condition)
    assert that this condition is true
```

Locators

```
new TreeItemLocator('General/Project')
    the tree item "Project" with parent "General"
new TreeItemLocator('(General|Simple)/Project')
    the tree item "Project" with parent "General" or "Simple"
new TreeItemLocator('Project',
    new ViewLocator('org.acme.Explorer'))
    the tree item "Project" in the "org.acme.Explorer" view
new MenuItemLocator('Window/&Preferences(...)?')
    the preferences menu (with or without trailing dots)
new CTabItemLocator('.*')
    all CTabItems
new XYLocator(new TableItemLocator('Java'), -5, 5)
    the offset -5,5 from the top-left of the "Java" table item
new LRLocator(3, new FigureClassLocator("Ellipse"))
    the 4th ellipse figure scanning left to right
```

Conditions

```
ui.wait(new ShellShowingCondition('New Project'))
    wait for the "New Project" shell to be showing
ui.wait(new ViewLocator('JUnit').isVisible())
    wait for the JUnit view to be visible
ui.wait(new ProjectExistsCondition('org.acme.MyProject'))
    wait for the "org.acme.MyProject" to exist
ui.wait(new JobsCompleteCondition())
    wait for the all the Eclipse Jobs to be complete
```

Assertions

```
ui.assertThat(new ButtonLocator('OK').isEnabled())
    assert that the "OK" button is enabled
ui.assertThat(new EditorLocator('Foo.java').isActive())
    assert that the editor open on "Foo.java" is active
ui.assertThat(
    new DialogMessageLocator().hasText('Error'))
    assert that the open dialog has the message "Error"
```

Condition Monitors

```
IShellMonitor sm =
    (IShellMonitor)ui.getAdapter(IShellMonitor.class)
sm.add(new PerspectiveSwitchedShellHandler())
    register a handler that clicks the "No" button on the "Con-
    firm Perspective Switch" Shell.
```

Text Entry

```
ui.enterText('MyProject')
    type "MyProject"
ui.keyClick(WT.CTRL, 's')
    click CTRL+S
ui.keyClick(WT.SHIFT | WT.END)
    click the SHIFT and END keys at once
```

Naming

```
button.setData('name', 'named.button')
    give the button the name "named.button"
ui.find(new NamedWidgetLocator('named.widget'))
    find the "named.button" button at runtime
```

Widget Access

```
IWidgetLocator wl = ui.find(new ButtonLocator('OK'))
    find the "OK" button
final Button b = (Button)((IWidgetReference)wl).getWidget()
    get the underlying widget
final boolean[] result = new boolean[1];
Display.getDefault().syncExec(new Runnable() {
    public void run() {
        result[0] = button.getAlignment();
    }
});
assertEquals(SWT.UP, result[0]);
    access the alignment property safely on the UI thread
```

Constants

```
WT.ARROW_RIGHT
    the right arrow key code
WT.setDefaultWaitTimeOut(120000)
    override the default wait timeout to be 2 minutes (in ms)
```

Drag and Drop

```
ui.click(new TreeItemLocator('accounts/due/PGE'));
ui.dragTo(new TreeItemLocator('accounts/paid'));
    drag "PGE" node to "accounts/paid"
ui.mouseMove(new AndGateLocator);
ui.dragTo(new FigureCanvasXYLocator(50, 50);
    drag the And gate to position (50,50) on the canvas
ui.click(new ResizeHandleLocator(Position.TOP,
    new FigureClassLocator("AndGate"));
ui.dragTo(new FigureCanvasXYLocator(60,10));
    resize the And gate figure using its TOP handle
```