Software-Entwicklung und Beratung

Jubula Reference Manual

BREDEX GmbH

October 12, 2014

BREDEX GmbH
Mauernstr. 33
38100 Braunschweig
Germany

Tel: +49-531 - 243 30 - 0
Fax: +49-531 - 243 30 - 99

www.bredex.de

Title:	Jubula Reference Manual
Author:	
File:	ReferenceManual
State:	RELEASE
Version:	V8.1.00171
Released by:	BREDEX GmbH
Released at:	October 12, 2014

Contents

Chapter 1

Introduction

This manual provides important information about the more technical side of working with the ITE.

Chapter 2

Components, Actions, and Parameters

This section provides a reference of all currently supported AUT components and the actions that can be applied to them.

There are two types of actions that can be applied to a given component: *execute* and *check*. *Execute* actions are marked with an 'e' in the description, whereas *check* actions are marked with a 'c'.

2.1 abstract Toolkit

The abstract toolkit contains component types which are available on all toolkits supported, and which have been specially adapted to be able to test various different components which share a set of features. These components are the *abstract components* available to make Test Steps more general.

2.1.1 Button Component

Description:

- This is an abstract component.
- It is implemented by all components which can be clicked like buttons.

Synopsis:

- Button Component (abstract)
  - Text Component (abstract)
    - Graphics Component (abstract)

New Actions

Name	Type	Parameters
Check Selection (→ page 19)	c	Boolean:Selected

Inherited Actions

Name	Type	Parameters	Inherited from
Check Enablement (→ page 23)	c	Boolean:Enabled	Graphics Component (abstract)
Check Enablement of Context Menu Entry by Indexpath (→ page 24)	c	String:Indexpath Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)
Check Enablement of Context Menu Entry by Indexpath (Specify Position) (→ page 25)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Indexpath Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)

Check Enablement of Context Menu Entry by Textpath (→ page 28)	c	String:Textpath String:Operator Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)
Check Enablement of Context Menu Entry by Textpath (Specify Position) (→ page 29)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Textpath String:Operator Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)
Check Existence (→ page 32)	c	Boolean:Exists	Graphics Component (abstract)
Check Existence of Context Menu Entry by Indexpath (→ page 33)	c	String:Indexpath Boolean:Exists Integer:Mouse Button	Graphics Component (abstract)
Check Existence of Context Menu Entry by Indexpath (Specify Position) (→ page 34)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Indexpath Boolean:Exists Integer:Mouse Button	Graphics Component (abstract)
Check Existence of Context Menu Entry by Textpath (→ page 37)	c	String:Textpath String:Operator Boolean:Exists Integer:Mouse Button	Graphics Component (abstract)
Check Existence of Context Menu Entry by Textpath (Specify Position) (→ page 38)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Textpath String:Operator Boolean:Exists Integer:Mouse Button	Graphics Component (abstract)

Check Focus (→ page 41)	c	Boolean:Has Focus	Graphics Component (abstract)
Check Property (→ page 42)	c	String:Property Name String:Property Value String:Operator	Graphics Component (abstract)
Check Selection of Context Menu Entry by Indexpath (→ page 43)	c	String:Indexpath Boolean:Selected Integer:Mouse Button	Graphics Component (abstract)
Check Selection of Context Menu Entry by Indexpath (Specify Position) (→ page 45)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Indexpath Boolean:Selected Integer:Mouse Button	Graphics Component (abstract)
Check Selection of Context Menu Entry by Textpath (→ page 47)	c	String:Textpath String:Operator Boolean:Selected Integer:Mouse Button	Graphics Component (abstract)
Check Selection of Context Menu Entry by Textpath (Specify Position) (→ page 49)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Textpath String:Operator Boolean:Selected Integer:Mouse Button	Graphics Component (abstract)

Check Text (→ page 76)	c	String:Text String:Operator	Text Component (abstract)
Click (→ page 52)	e	Integer:Number of Clicks Integer:Mouse Button	Graphics Component (abstract)
Click in Component (→ page 53)	e	Integer:Number of Clicks Integer:Mouse Button Integer:x-position String:x-units Integer:y-position String:y-units	Graphics Component (abstract)
Drag (→ page 55)	e	Integer:Mouse Button String:Modifier Keys Integer:x-position String:x-units Integer:y-position String:y-units	Graphics Component (abstract)
Drop (→ page 58)	e	Integer:x-position String:x-units Integer:y-position String:y-units Integer:Delay before drop (milliseconds)	Graphics Component (abstract)

Select Context Menu Entry by Indexpath (→ page 60)	e	String:Indexpath Integer:Mouse Button	Graphics Component (abstract)
Select Context Menu Entry by Indexpath (Specify Position) (→ page 61)	e	Integer:x-position String:x-units Integer:y-position String:y-units String:Indexpath Integer:Mouse Button	Graphics Component (abstract)
Select Context Menu Entry by Textpath (→ page 64)	e	String:Textpath String:Operator Integer:Mouse Button	Graphics Component (abstract)
Select Context Menu Entry by Textpath (Specify Position) (→ page 66)	e	Integer:x-position String:x-units Integer:y-position String:y-units String:Menupath String:Operator Integer:Mouse Button	Graphics Component (abstract)
Show Text (→ page 69)	e	String:Text Integer:Text Size (in points) Integer:Time per Word (in milliseconds) Integer:Window Width (in pixels)	Graphics Component (abstract)
Store Property (→ page 70)	e	Variable:Variable Name String:Property Name	Graphics Component (abstract)
Store Value (→ page 77)	e	Variable:Variable Name	Text Component (abstract)

Wait for Component (→ page 71)	e	Integer:Timeout in ms Integer:Delay after Visibility	Graphics Component (abstract)

2.1.1.1 Check Selection

`(Button Component)`

- Use this action to check if the button is currently selected or not.
- This action is particularly useful for check boxes and radio buttons.

Parameters

Name	Data Type	Values	Default
Selected	Boolean	true false	true

- Set this parameter to true if you expect the button to be selected.
- Set this parameter to false if you expect the button to *not* be selected.

Used By

(SWT) Tool Item (swt)
Button Component (concrete)

2.1.2 Graphics Component

Description:

- This is an abstract component.
- It represents the basic properties of all of the objects in an application, for example:
  - if they are enabled (manipulable)
  - if they are selected (have focus)
  - other properties (e.g. size, shape, color)

Synopsis:

- Graphics Component (abstract)

New Actions

Name	Type	Parameters
Check Enablement (→ page 23)	c	Boolean:Enabled
Check Enablement of Context Menu Entry by Indexpath (→ page 24)	c	String:Indexpath Boolean:Enabled Integer:Mouse Button
Check Enablement of Context Menu Entry by Indexpath (Specify Position) (→ page 25)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Indexpath Boolean:Enabled Integer:Mouse Button
Check Enablement of Context Menu Entry by Textpath (→ page 28)	c	String:Textpath String:Operator Boolean:Enabled Integer:Mouse Button
Check Enablement of Context Menu Entry by Textpath (Specify Position) (→ page 29)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Textpath String:Operator Boolean:Enabled Integer:Mouse Button

Check Existence (→ page 32)	c	Boolean:Exists
Check Existence of Context Menu Entry by Indexpath (→ page 33)	c	String:Indexpath Boolean:Exists Integer:Mouse Button
Check Existence of Context Menu Entry by Indexpath (Specify Position) (→ page 34)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Indexpath Boolean:Exists Integer:Mouse Button
Check Existence of Context Menu Entry by Textpath (→ page 37)	c	String:Textpath String:Operator Boolean:Exists Integer:Mouse Button
Check Existence of Context Menu Entry by Textpath (Specify Position) (→ page 38)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Textpath String:Operator Boolean:Exists Integer:Mouse Button
Check Focus (→ page 41)	c	Boolean:Has Focus
Check Property (→ page 42)	c	String:Property Name String:Property Value String:Operator
Check Selection of Context Menu Entry by Indexpath (→ page 43)	c	String:Indexpath Boolean:Selected Integer:Mouse Button
Check Selection of Context Menu Entry by Indexpath (Specify Position) (→ page 45)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Indexpath Boolean:Selected Integer:Mouse Button

Check Selection of Context Menu Entry by Textpath (→ page 47)	c	String:Textpath String:Operator Boolean:Selected Integer:Mouse Button
Check Selection of Context Menu Entry by Textpath (Specify Position) (→ page 49)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Textpath String:Operator Boolean:Selected Integer:Mouse Button
Click (→ page 52)	e	Integer:Number of Clicks Integer:Mouse Button
Click in Component (→ page 53)	e	Integer:Number of Clicks Integer:Mouse Button Integer:x-position String:x-units Integer:y-position String:y-units
Drag (→ page 55)	e	Integer:Mouse Button String:Modifier Keys Integer:x-position String:x-units Integer:y-position String:y-units
Drop (→ page 58)	e	Integer:x-position String:x-units Integer:y-position String:y-units Integer:Delay before drop (milliseconds)
Select Context Menu Entry by Indexpath (→ page 60)	e	String:Indexpath Integer:Mouse Button
Select Context Menu Entry by Indexpath (Specify Position) (→ page 61)	e	Integer:x-position String:x-units Integer:y-position String:y-units String:Indexpath Integer:Mouse Button

Select Context Menu Entry by Textpath (→ page 64)	e	String:Textpath String:Operator Integer:Mouse Button
Select Context Menu Entry by Textpath (Specify Position) (→ page 66)	e	Integer:x-position String:x-units Integer:y-position String:y-units String:Menupath String:Operator Integer:Mouse Button
Show Text (→ page 69)	e	String:Text Integer:Text Size (in points) Integer:Time per Word (in milliseconds) Integer:Window Width (in pixels)
Store Property (→ page 70)	e	Variable:Variable Name String:Property Name
Wait for Component (→ page 71)	e	Integer:Timeout in ms Integer:Delay after Visibility

Inherited Actions

none

2.1.2.1 Check Enablement

`(Graphics Component)`

- Use this action to check if the graphics component is active (manipulable) within the application.

  Parameters

Name	Data Type	Values	Default
Enabled	Boolean	true false	true

  - Set this parameter to true if you expect the component to be enabled.
  - Set this parameter to false if you expect the component to be disabled.

2.1.2.2 Check Enablement of Context Menu Entry by Indexpath

`(Graphics Component)`

- Use this action to check the enablement status of an item in a context menu.
- The item to check is given using the indexpath to the item.

This action is unsupported for the HTML toolkit

Parameters

Name	Data Type	Values	Default
Indexpath	String	–	none

- Enter the path to the item as an indexpath.
- Use slash `'/'` as a path separator (to separate parent nodes from child nodes).
- For example, `1/2` (without quotes).

The first node is '1' (without quotes)

Name	Data Type	Values	Default
Enabled	Boolean	true false	true

- Set this parameter to true if you expect the menu item to be enabled.
- Set the parameter to false if you expect the menu item to be disabled.

Name	Data Type	Values	Default
Mouse Button	Integer	1 2 3	3

- Use this parameter to select which mouse button you want to click with.
  - `1` = left mouse button
  - `2` = middle mouse button
  - `3` = right mouse button

- The mouse button you use to click with can also have an effect on what happens.
- The button to click will depend on your AUT and what the click should achieve.

Do not enter quotes around the mouse button numbers, en them in plain text.

2.1.2.3 Check Enablement of Context Menu Entry by Indexpath (Specify Position)

`(Graphics Component)`

- Use this action to check the enablement status of an item in a context menu.
- The item to check is given using the indexpath to the item.
- The context menu is opened at a specific place in the component, specified by the x- and y- coordinates.
- This can sometimes produce a different context menu than when you click on a specific node/cell etc.
- If you want to check a context-sensitive menu on a specific node or cell in a component, then use the action which checks the context menu without specifying the coordinate position.
- You can also define which mouse button should be used to open the context menu.

This action is unsupported for the HTML toolkit

Parameters

Name	Data Type	Values	Default
x-position	Integer	–	50

- With this parameter, you can specify where in the component you should click.
- Use this parameter to enter a value for the x-axis (the horizontal position).

- Using the x-units parameter, you can specify whether the value should be in pixels or in percent.
- For example, if you enter `50` and then `percent` (without quotes) , then the click will be performed in the middle of the x-axis.
- All positions are calculated from the upper left of the component, with the positive x-axis stretching toward the right side of the component.

We recommend not using `0` (pixels and percent) or `100`(percent) as positions.

Name	Data Type	Values	Default
x-units	String	percent pixel	percent

- Use this parameter to specify whether you want your values for the x-coordinate to be in pixels or percent.
- If you choose percent for this parameter, and `50` for the *x-position* parameter, the click will always be in the middle of the x-axis.

Name	Data Type	Values	Default
y-position	Integer	–	50

- With this parameter, you can specify where in the component you should click.
- Use this parameter to enter a value for the y-axis (the vertical position).
- Using the y-units parameter, you can specify whether the value should be in pixels or in percent.
- For example, if you enter `50` and then `percent` (without quotes), then the click will be performed in the middle of the y-axis.
- All positions are calculated from the upper left of the component, with the positive y-axis stretching toward the bottom of the component.

We recommend not using `0` (pixels and percent) or `100`(percent) as positions.

Name	Data Type	Values	Default
y-units	String	percent pixel	percent

- Use this parameter to specify whether you want your values for the y-coordinate to be in pixels or percent.

- If you choose percent for this parameter, and `50` for the *y-position* parameter, the click will always be in the middle of the y-axis.

Name	Data Type	Values	Default
Indexpath	String	–	none

- Enter the path to the item as an indexpath.
- Use slash `'/'` as a path separator (to separate parent nodes from child nodes).
- For example, `1/2` (without quotes).

The first node is '1' (without quotes)

Name	Data Type	Values	Default
Enabled	Boolean	true false	true

- Set this parameter to true if you expect the item to be enabled.
- Set this parameter to false if you expect the item to be disabled.

Name	Data Type	Values	Default
Mouse Button	Integer	1 2 3	3

- Use this parameter to select which mouse button you want to click with.
  - `1` = left mouse button
  - `2` = middle mouse button
  - `3` = right mouse button
- The mouse button you use to click with can also have an effect on what happens.
- The button to click will depend on your AUT and what the click should achieve.

Do not enter quotes around the mouse button numbers, en them in plain text.

2.1.2.4 Check Enablement of Context Menu Entry by Textpath

`(Graphics Component)`

- Use this action to check whether an item in a context menu is enabled.
- The item to check is given using the textpath to the item.

This action is unsupported for the HTML toolkit

Parameters

Name	Data Type	Values	Default
Textpath	String	–	none

- Enter the path to the item as a textpath.
- Use slash `'/'` as a path separator (to separate parent nodes from child nodes).
- For example, `File/Open` or `Category/Horror` (without quotes).
- Either make sure that your path is written exactly as it appears in the interface, or use a regular expression to match the text.
- Each segment of the path will be used to find a corresponding node, using the operator provided.

Name	Data Type	Values	Default
Operator	String	equals not equals matches simple match	equals

- The *operator* parameter has four possible values.
- "*not equals*" looks for something that does *not exactly* match.
- "*equals*" looks for an *exact* match.
- Select "*simple match*" to use a simple match expression (→ page **??**) .
- Select "*matches*" to use a regular expression (→ page **??**) .
- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

Name	Data Type	Values	Default
Enabled	Boolean	true false	true

- Set this parameter to true if you expect the menu item to be enabled.
- Set the parameter to false if you expect the menu item to be disabled.

Name	Data Type	Values	Default
Mouse Button	Integer	1 2 3	3

- Use this parameter to select which mouse button you want to click with.
  - `1` = left mouse button
  - `2` = middle mouse button
  - `3` = right mouse button
- The mouse button you use to click with can also have an effect on what happens.
- The button to click will depend on your AUT and what the click should achieve.

Do not enter quotes around the mouse button numbers, en them in plain text.

2.1.2.5 Check Enablement of Context Menu Entry by Textpath (Specify Position)

`(Graphics Component)`

- Use this action to check whether an item in a context menu is enabled.
- The item to check is given using the textpath to the item.
- This action will open the context menu at the position you specify in the component.
- For some components (e.g. trees), opening a context menu somewhere else in the component can have a different effect than when you open the context menu on a specific node.

- If you want to open or check a context menu on a selected node or cell in a component, then use the action which selects the context menu without specifying the coordinate position.

This action is unsupported for the HTML toolkit

Parameters

Name	Data Type	Values	Default
x-position	Integer	–	50

- With this parameter, you can specify where in the component you should click.
- Use this parameter to enter a value for the x-axis (the horizontal position).
- Using the x-units parameter, you can specify whether the value should be in pixels or in percent.
- For example, if you enter `50` and then `percent` (without quotes) , then the click will be performed in the middle of the x-axis.
- All positions are calculated from the upper left of the component, with the positive x-axis stretching toward the right side of the component.

We recommend not using `0` (pixels and percent) or `100`(percent) as positions.

Name	Data Type	Values	Default
x-units	String	percent pixel	percent

- Use this parameter to specify whether you want your values for the x-coordinate to be in pixels or percent.
- If you choose percent for this parameter, and `50` for the *x-position* parameter, the click will always be in the middle of the x-axis.

Name	Data Type	Values	Default
y-position	Integer	–	50

- With this parameter, you can specify where in the component you should click.
- Use this parameter to enter a value for the y-axis (the vertical position).

- Using the y-units parameter, you can specify whether the value should be in pixels or in percent.
- For example, if you enter `50` and then `percent` (without quotes), then the click will be performed in the middle of the y-axis.
- All positions are calculated from the upper left of the component, with the positive y-axis stretching toward the bottom of the component.

We recommend not using `0` (pixels and percent) or `100`(percent) positions.

Name	Data Type	Values	Default
y-units	String	percent pixel	percent

- Use this parameter to specify whether you want your values for the y-coordinate to be in pixels or percent.
- If you choose percent for this parameter, and `50` for the *y-position* parameter, the click will always be in the middle of the y-axis.

Name	Data Type	Values	Default
Textpath	String	–	none

- Enter the path to the item as a textpath.
- Use slash `'/'` as a path separator (to separate parent nodes from child nodes).
- For example, `File/Open` or `Category/Horror` (without quotes).
- Either make sure that your path is written exactly as it appears in the interface, or use a regular expression to match the text.
- Each segment of the path will be used to find a corresponding node, using the operator provided.

Name	Data Type	Values	Default
Operator	String	equals not equals matches simple match	equals

- The *operator* parameter has four possible values.
- "*not equals*" looks for something that does *not exactly* match.
- "*equals*" looks for an *exact* match.
- Select "*simple match*" to use a simple match expression (→ page **??**) .

- Select "*matches*" to use a regular expression (→ page **??**) .
- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

Name	Data Type	Values	Default
Enabled	Boolean	true false	true

- Set this parameter to true if you expect the context menu item to be enabled.
- Set the parameter to false if you expect the context menu item to be disabled.

Name	Data Type	Values	Default
Mouse Button	Integer	1 2 3	3

- Use this parameter to select which mouse button you want to click with.
  - `1` = left mouse button
  - `2` = middle mouse button
  - `3` = right mouse button
- The mouse button you use to click with can also have an effect on what happens.
- The button to click will depend on your AUT and what the click should achieve.

Do not enter quotes around the mouse button numbers, enter them in plain text.

2.1.2.6 Check Existence

`(Graphics Component)`

- Use this action to check the existence of a component within the application.

Parameters

Name	Data Type	Values	Default
Exists	Boolean	true false	true

- Set this parameter to true if you expect the component to exist.
- Set this parameter to false if you do *not* expect the component to exist.

2.1.2.7 Check Existence of Context Menu Entry by Indexpath

`(Graphics Component)`

- Use this action to check whether an item in a context menu exists.
- The item to check is given using the indexpath to the item.

This action is unsupported for the HTML toolkit

Parameters

Name	Data Type	Values	Default
Indexpath	String	–	none

- Enter the path to the item as an indexpath.
- Use slash `'/'` as a path separator (to separate parent nodes from child nodes).
- For example, `1/2` (without quotes).

The first node is '1' (without quotes)

Name	Data Type	Values	Default
Exists	Boolean	true false	true

- Set this parameter to true if you expect the context menu item to exist.
- Set the parameter to false if you expect the context menu item to *not* exist.

Name	Data Type	Values	Default
Mouse Button	Integer	1 2 3	3

- Use this parameter to select which mouse button you want to click with.
  - `1` = left mouse button
  - `2` = middle mouse button
  - `3` = right mouse button
- The mouse button you use to click with can also have an effect on what happens.
- The button to click will depend on your AUT and what the click should achieve.

Do not enter quotes around the mouse button numbers, enter them in plain text.

2.1.2.8 Check Existence of Context Menu Entry by Indexpath (Specify Position)

`(Graphics Component)`

- Use this action to check whether an item in a context menu exists.
- The item to check is given using the indexpath to the item.
- This action will open the context menu at the position you specify in the component.
- For some components (e.g. trees), opening a context menu somewhere else in the component can have a different effect than when you open the context menu on a specific node.
- If you want to open or check a context menu on a selected node or cell in a component, then use the action which selects the context menu without specifying the coordinate position.

This action is unsupported for the HTML toolkit

Parameters

Name	Data Type	Values	Default
x-position	Integer	–	50

- With this parameter, you can specify where in the component you should click.
- Use this parameter to enter a value for the x-axis (the horizontal position).
- Using the x-units parameter, you can specify whether the value should be in pixels or in percent.
- For example, if you enter `50` and then `percent` (without quotes) , then the click will be performed in the middle of the x-axis.
- All positions are calculated from the upper left of the component, with the positive x-axis stretching toward the right side of the component.

We recommend not using `0` (pixels and percent) or `100`(percent) positions.

Name	Data Type	Values	Default
x-units	String	percent pixel	percent

- Use this parameter to specify whether you want your values for the x-coordinate to be in pixels or percent.
- If you choose percent for this parameter, and `50` for the *x-position* parameter, the click will always be in the middle of the x-axis.

Name	Data Type	Values	Default
y-position	Integer	–	50

- With this parameter, you can specify where in the component you should click.
- Use this parameter to enter a value for the y-axis (the vertical position).
- Using the y-units parameter, you can specify whether the value should be in pixels or in percent.
- For example, if you enter `50` and then `percent` (without quotes), then the click will be performed in the middle of the y-axis.
- All positions are calculated from the upper left of the component, with the positive y-axis stretching toward the bottom of the component.

We recommend not using `0` (pixels and percent) or `100`(percent) positions.

Name	Data Type	Values	Default
y-units	String	percent pixel	percent

- Use this parameter to specify whether you want your values for the y-coordinate to be in pixels or percent.
- If you choose percent for this parameter, and `50` for the *y-position* parameter, the click will always be in the middle of the y-axis.

Name	Data Type	Values	Default
Indexpath	String	–	none

- Enter the path to the item as an indexpath.
- Use slash `'/'` as a path separator (to separate parent nodes from child nodes).
- For example, `1/2` (without quotes).

The first node is ’1’ (without quotes)

Name	Data Type	Values	Default
Exists	Boolean	true false	true

- Set this parameter to true if you expect the context menu item to exist.
- Set the parameter to false if you expect the context menu item to *not* exist.

Name	Data Type	Values	Default
Mouse Button	Integer	1 2 3	3

- Use this parameter to select which mouse button you want to click with.
  - `1` = left mouse button
  - `2` = middle mouse button
  - `3` = right mouse button
- The mouse button you use to click with can also have an effect on what happens.
- The button to click will depend on your AUT and what the click should achieve.

Do not enter quotes around the mouse button numbers, enter them in plain text.

2.1.2.9 Check Existence of Context Menu Entry by Textpath

`(Graphics Component)`

- Use this action to check the existence of an item in a context menu.
- The item to check is given using the textpath to the item.

This action is unsupported for the HTML toolkit

Parameters

Name	Data Type	Values	Default
Textpath	String	–	none

- Enter the path to the item as an indexpath.
- Use slash `'/'` as a path separator (to separate parent nodes from child nodes).
- For example, `1/2` (without quotes).

The first node is '1' (without quotes)

Name	Data Type	Values	Default
Operator	String	equals not equals matches simple match	equals

- The *operator* parameter has four possible values.
- "*not equals*" looks for something that does *not exactly* match.
- "*equals*" looks for an *exact* match.
- Select "*simple match*" to use a simple match expression (→ page **??**) .
- Select "*matches*" to use a regular expression (→ page **??**) .
- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

Name	Data Type	Values	Default
Exists	Boolean	true false	true

- Set this parameter to true if you expect the context menu item to exist.
- Set the parameter to false if you expect the context menu item to *not* exist.

Name	Data Type	Values	Default
Mouse Button	Integer	1 2 3	3

- Use this parameter to select which mouse button you want to click with.
  - `1` = left mouse button
  - `2` = middle mouse button
  - `3` = right mouse button
- The mouse button you use to click with can also have an effect on what happens.
- The button to click will depend on your AUT and what the click should achieve.

Do not enter quotes around the mouse button numbers, enter them in plain text.

2.1.2.10 Check Existence of Context Menu Entry by Textpath (Specify Position)

`(Graphics Component)`

- Use this action to check whether an item in a context menu exists.
- The item to check is given using the textpath to the item.
- This action will open the context menu at the position you specify in the component.
- For some components (e.g. trees), opening a context menu somewhere else in the component can have a different effect than when you open the context menu on a specific node.

- If you want to open or check a context menu on a selected node or cell in a component, then use the action which selects the context menu without specifying the coordinate position.

This action is unsupported for the HTML toolkit

Parameters

Name	Data Type	Values	Default
x-position	Integer	–	50

- With this parameter, you can specify where in the component you should click.
- Use this parameter to enter a value for the x-axis (the horizontal position).
- Using the x-units parameter, you can specify whether the value should be in pixels or in percent.
- For example, if you enter `50` and then `percent` (without quotes) , then the click will be performed in the middle of the x-axis.
- All positions are calculated from the upper left of the component, with the positive x-axis stretching toward the right side of the component.

We recommend not using `0` (pixels and percent) or `100`(percent) positions.

Name	Data Type	Values	Default
x-units	String	percent pixel	percent

- Use this parameter to specify whether you want your values for the x-coordinate to be in pixels or percent.
- If you choose percent for this parameter, and `50` for the *x-position* parameter, the click will always be in the middle of the x-axis.

Name	Data Type	Values	Default
y-position	Integer	–	50

- With this parameter, you can specify where in the component you should click.
- Use this parameter to enter a value for the y-axis (the vertical position).

- Using the y-units parameter, you can specify whether the value should be in pixels or in percent.
- For example, if you enter `50` and then `percent` (without quotes), then the click will be performed in the middle of the y-axis.
- All positions are calculated from the upper left of the component, with the positive y-axis stretching toward the bottom of the component.

We recommend not using `0` (pixels and percent) or `100`(percent) as positions.

Name	Data Type	Values	Default
y-units	String	percent pixel	percent

- Use this parameter to specify whether you want your values for the y-coordinate to be in pixels or percent.
- If you choose percent for this parameter, and `50` for the *y-position* parameter, the click will always be in the middle of the y-axis.

Name	Data Type	Values	Default
Textpath	String	–	none

- Enter the path to the item as a textpath.
- Use slash `'/'` as a path separator (to separate parent nodes from child nodes).
- For example, `File/Open` or `Category/Horror` (without quotes).
- Either make sure that your path is written exactly as it appears in the interface, or use a regular expression to match the text.
- Each segment of the path will be used to find a corresponding node, using the operator provided.

Name	Data Type	Values	Default
Operator	String	equals not equals matches simple match	equals

- The *operator* parameter has four possible values.
- "*not equals*" looks for something that does *not exactly* match.
- "*equals*" looks for an *exact* match.
- Select "*simple match*" to use a simple match expression (→ page **??**) .

- Select "*matches*" to use a regular expression (→ page **??**) .
- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

Name	Data Type	Values	Default
Exists	Boolean	true false	true

- Set this parameter to true if you expect the context menu item to exist.
- Set the parameter to false if you expect the context menu item to *not* exist.

Name	Data Type	Values	Default
Mouse Button	Integer	1 2 3	3

- Use this parameter to select which mouse button you want to click with.
  - `1` = left mouse button
  - `2` = middle mouse button
  - `3` = right mouse button
- The mouse button you use to click with can also have an effect on what happens.
- The button to click will depend on your AUT and what the click should achieve.

Do not enter quotes around the mouse button numbers, en them in plain text.

2.1.2.11 Check Focus

`(Graphics Component)`

- Use this action to check whether the graphics component currently has the focus.
- When a component has focus, it can receive input.

This action is unsupported for the HTML toolkit

Parameters

Name	Data Type	Values	Default
Has Focus	Boolean	true false	true

- Set this parameter to true if you expect the component to be in focus.
- Set this parameter to false if you expect the component *not* to be in focus.

2.1.2.12 Check Property

`(Graphics Component)`

- Use this action to check a specific property of a component.
- You can check whether a value you enter for a given property matches the actual value for the property.
- Every component has properties (attributes). These can include the text size, the size (width/height) of the component, the color etc
- You must first find out the name of the property you want to check, and then what value it should have. You can find out what properties and values a component has in the Object Mapping Mode (for Swing and SWT/RCP AUT's). When you collect a component, its property information is displayed in the Properties View.

If the property is *private*, it cannot be checked, as the remote control component has no access to it.

When testing HTML AUT's, there are some differences between the properties available in Internet Explorer and Firefox. Some properties available in FF may not be retrievable under IE.

Parameters

Name	Data Type	Values	Default
Property Name	String	–	none

- Enter the name of the property you want to check.
- For example, to check the width, enter `width`.

Name	Data Type	Values	Default
Property Value	String	–	none

- Enter the value you expect the property to have.
- For `width` and `height` properties, enter the value in pixels.

Name	Data Type	Values	Default
Operator	String	equals not equals matches simple match	equals

- The *operator* parameter has four possible values.
- "*not equals*" looks for something that does *not exactly* match.
- "*equals*" looks for an *exact* match.
- Select "*simple match*" to use a simple match expression (→ page **??**) .
- Select "*matches*" to use a regular expression (→ page **??**) .
- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

2.1.2.13 Check Selection of Context Menu Entry by Indexpath

`(Graphics Component)`

- Use this action to check whether an item in a context menu is selected.

- You enter the menupath to the item, and whether you expect it to be selected or not.
- The menupath is given as an indexpath.

This action is unsupported for the HTML toolkit

Parameters

Name	Data Type	Values	Default
Indexpath	String	–	none

- Enter the path to the item as an indexpath.
- Use slash `'/'` as a path separator (to separate parent nodes from child nodes).
- For example, `1/2` (without quotes).

The first node is '1' (without quotes)

Name	Data Type	Values	Default
Selected	Boolean	true false	true

- Set this parameter to true if you expect the context menu item to be selected.
- Set this parameter to false if you expect the context menu item *not* to be selected.

Name	Data Type	Values	Default
Mouse Button	Integer	1 2 3	3

- Use this parameter to select which mouse button you want to click with.
  - `1` = left mouse button
  - `2` = middle mouse button
  - `3` = right mouse button
- The mouse button you use to click with can also have an effect on what happens.
- The button to click will depend on your AUT and what the click should achieve.

Do not enter quotes around the mouse button numbers, enter them in plain text.

2.1.2.14 Check Selection of Context Menu Entry by Indexpath (Specify Position)

`(Graphics Component)`

- Use this action to check whether an item in a context menu is selected.
- The item to check is given using the indexpath to the item.
- This action will open the context menu at the position you specify in the component.
- For some components (e.g. trees), opening a context menu somewhere else in the component can have a different effect than when you open the context menu on a specific node.
- If you want to open or check a context menu on a selected node or cell in a component, then use the action which selects the context menu without specifying the coordinate position.

This action is unsupported for the HTML toolkit

Parameters

Name	Data Type	Values	Default
x-position	Integer	–	50

- With this parameter, you can specify where in the component you should click.
- Use this parameter to enter a value for the x-axis (the horizontal position).
- Using the x-units parameter, you can specify whether the value should be in pixels or in percent.
- For example, if you enter `50` and then `percent` (without quotes) , then the click will be performed in the middle of the x-axis.
- All positions are calculated from the upper left of the component, with the positive x-axis stretching toward the right side of the component.

We recommend not using `0` (pixels and percent) or `100`(percent) positions.

Name	Data Type	Values	Default
x-units	String	percent pixel	percent

- Use this parameter to specify whether you want your values for the x-coordinate to be in pixels or percent.
- If you choose percent for this parameter, and `50` for the *x-position* parameter, the click will always be in the middle of the x-axis.

Name	Data Type	Values	Default
y-position	Integer	–	50

- With this parameter, you can specify where in the component you should click.
- Use this parameter to enter a value for the y-axis (the vertical position).
- Using the y-units parameter, you can specify whether the value should be in pixels or in percent.
- For example, if you enter `50` and then `percent` (without quotes), then the click will be performed in the middle of the y-axis.
- All positions are calculated from the upper left of the component, with the positive y-axis stretching toward the bottom of the component.

We recommend not using `0` (pixels and percent) or `100`(percent) as positions.

Name	Data Type	Values	Default
y-units	String	percent pixel	percent

- Use this parameter to specify whether you want your values for the y-coordinate to be in pixels or percent.
- If you choose percent for this parameter, and `50` for the *y-position* parameter, the click will always be in the middle of the y-axis.

Name	Data Type	Values	Default
Indexpath	String	–	none

- Enter the path to the item as an indexpath.
- Use slash `'/'` as a path separator (to separate parent nodes from child nodes).

- For example, `1/2` (without quotes).

The first node is ’1’ (without quotes)

Name	Data Type	Values	Default
Selected	Boolean	true false	true

- Set this parameter to true if you expect the context menu item to be selected.
- Set this parameter to false if you expect the context menu item *not* to be selected.

Name	Data Type	Values	Default
Mouse Button	Integer	1 2 3	3

- Use this parameter to select which mouse button you want to click with.
  - `1` = left mouse button
  - `2` = middle mouse button
  - `3` = right mouse button
- The mouse button you use to click with can also have an effect on what happens.
- The button to click will depend on your AUT and what the click should achieve.

Do not enter quotes around the mouse button numbers, en them in plain text.

2.1.2.15 Check Selection of Context Menu Entry by Textpath

`(Graphics Component)`

- Use this action to check whether an item in a context menu is selected.
- You enter the menupath to the item, and whether you expect it to be selected or not.

- The menupath is given as a textpath.

This action is unsupported for the HTML toolkit

Parameters

Name	Data Type	Values	Default
Textpath	String	–	none

- Enter the path to the item as a textpath.
- Use slash `'/'` as a path separator (to separate parent nodes from child nodes).
- For example, `File/Open` or `Category/Horror` (without quotes).
- Either make sure that your path is written exactly as it appears in the interface, or use a regular expression to match the text.
- Each segment of the path will be used to find a corresponding node, using the operator provided.

Name	Data Type	Values	Default
Operator	String	equals not equals matches simple match	equals

- The *operator* parameter has four possible values.
- "*not equals*" looks for something that does *not exactly* match.
- "*equals*" looks for an *exact* match.
- Select "*simple match*" to use a simple match expression (→ page **??**) .
- Select "*matches*" to use a regular expression (→ page **??**) .
- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

Name	Data Type	Values	Default
Selected	Boolean	true false	true

- Set this parameter to true if you expect the context menu item to be selected.
- Set this parameter to false if you expect the context menu item *not* to be selected.

Name	Data Type	Values	Default
Mouse Button	Integer	1 2 3	3

- Use this parameter to select which mouse button you want to click with.
  - 1 = left mouse button
  - 2 = middle mouse button
  - 3 = right mouse button
- The mouse button you use to click with can also have an effect on what happens.
- The button to click will depend on your AUT and what the click should achieve.

Do not enter quotes around the mouse button numbers, en them in plain text.

2.1.2.16 Check Selection of Context Menu Entry by Textpath (Specify Position)

`(Graphics Component)`

- Use this action to check whether an item in a context menu is selected.
- The item to check is given using the textpath to the item.
- This action will open the context menu at the position you specify in the component.
- For some components (e.g. trees), opening a context menu somewhere else in the component can have a different effect than when you open the context menu on a specific node.
- If you want to open or check a context menu on a selected node or cell in a component, then use the action which selects the context menu without specifying the coordinate position.

This action is unsupported for the HTML toolkit

Parameters

Name	Data Type	Values	Default
x-position	Integer	–	50

- With this parameter, you can specify where in the component you should click.
- Use this parameter to enter a value for the x-axis (the horizontal position).
- Using the x-units parameter, you can specify whether the value should be in pixels or in percent.
- For example, if you enter `50` and then `percent` (without quotes) , then the click will be performed in the middle of the x-axis.
- All positions are calculated from the upper left of the component, with the positive x-axis stretching toward the right side of the component.

We recommend not using `0` (pixels and percent) or `100`(percent) as positions.

Name	Data Type	Values	Default
x-units	String	percent pixel	percent

- Use this parameter to specify whether you want your values for the x-coordinate to be in pixels or percent.
- If you choose percent for this parameter, and `50` for the *x-position* parameter, the click will always be in the middle of the x-axis.

Name	Data Type	Values	Default
y-position	Integer	–	50

- With this parameter, you can specify where in the component you should click.
- Use this parameter to enter a value for the y-axis (the vertical position).
- Using the y-units parameter, you can specify whether the value should be in pixels or in percent.
- For example, if you enter `50` and then `percent` (without quotes), then the click will be performed in the middle of the y-axis.
- All positions are calculated from the upper left of the component, with the positive y-axis stretching toward the bottom of the component.

We recommend not using `0` (pixels and percent) or `100`(percent) as positions.

Name	Data Type	Values	Default
y-units	String	percent pixel	percent

- Use this parameter to specify whether you want your values for the y-coordinate to be in pixels or percent.
- If you choose percent for this parameter, and `50` for the *y-position* parameter, the click will always be in the middle of the y-axis.

Name	Data Type	Values	Default
Textpath	String	–	none

- Enter the path to the item as a textpath.
- Use slash `'/'` as a path separator (to separate parent nodes from child nodes).
- For example, `File/Open` or `Category/Horror` (without quotes).
- Either make sure that your path is written exactly as it appears in the interface, or use a regular expression to match the text.
- Each segment of the path will be used to find a corresponding node, using the operator provided.

Name	Data Type	Values	Default
Operator	String	equals not equals matches simple match	equals

- The *operator* parameter has four possible values.
- "*not equals*" looks for something that does *not exactly* match.
- "*equals*" looks for an *exact* match.
- Select "*simple match*" to use a simple match expression (→ page **??**) .
- Select "*matches*" to use a regular expression (→ page **??**) .
- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

Name	Data Type	Values	Default
Selected	Boolean	true false	true

- Set this parameter to true if you expect the context menu item to be selected.
- Set this parameter to false if you expect the context menu item *not* to be selected.

Name	Data Type	Values	Default
Mouse Button	Integer	1 2 3	3

- Use this parameter to select which mouse button you want to click with.
  - `1` = left mouse button
  - `2` = middle mouse button
  - `3` = right mouse button
- The mouse button you use to click with can also have an effect on what happens.
- The button to click will depend on your AUT and what the click should achieve.

Do not enter quotes around the mouse button numbers, enter them in plain text.

2.1.2.17 Click

`(Graphics Component)`

- This action sends a "click" to the component.
- You can click once or multiple times.
- You can also specify which mouse button (left, middle, right) you want to click with.

Parameters

Name	Data Type	Values	Default
Number of Clicks	Integer	–	1

- Select how many clicks you want to make on the component.
- Different click counts can cause different results.
- For example:
  - A single click on an editable text field puts the cursor in the text field
  - A triple click on an editable text field selects the whole text.
- Depending on your application, you may need to send one, two or three clicks to a component.

Name	Data Type	Values	Default
Mouse Button	Integer	1 2 3	1

- Use this parameter to select which mouse button you want to click with.
  - `1` = left mouse button
  - `2` = middle mouse button
  - `3` = right mouse button
- The mouse button you use to click with can also have an effect on what happens.
- The button to click will depend on your AUT and what the click should achieve.

Do not enter quotes around the mouse button numbers, en them in plain text.

2.1.2.18 Click in Component

`(Graphics Component)`

- This item sends one or more mouse clicks to the component.
- This is realized by:
  - Clicking the specified mouse button the given number of times at the location indicated.

Parameters

Name	Data Type	Values	Default
Number of Clicks	Integer	–	1

- Select how many clicks you want to make on the component.
- Different click counts can cause different results.
- For example:
  - A single click on an editable text field puts the cursor in the text field
  - A triple click on an editable text field selects the whole text.
- Depending on your application, you may need to send one, two or three clicks to a component.

Name	Data Type	Values	Default
Mouse Button	Integer	1 2 3	1

- Use this parameter to select which mouse button you want to click with.
  - `1` = left mouse button
  - `2` = middle mouse button
  - `3` = right mouse button
- The mouse button you use to click with can also have an effect on what happens.
- The button to click will depend on your AUT and what the click should achieve.

Do not enter quotes around the mouse button numbers, enter them in plain text.

Name	Data Type	Values	Default
x-position	Integer	–	50

- With this parameter, you can specify where in the component you should click.
- Use this parameter to enter a value for the x-axis (the horizontal position).
- Using the x-units parameter, you can specify whether the value should be in pixels or in percent.
- For example, if you enter `50` and then `percent` (without quotes) , then the click will be performed in the middle of the x-axis.
- All positions are calculated from the upper left of the component, with the positive x-axis stretching toward the right side of the component.

We recommend not using `0` (pixels and percent) or `100`(percent) as positions.

Name	Data Type	Values	Default
x-units	String	percent pixel	percent

- Use this parameter to specify whether you want your values for the x-coordinate to be in pixels or percent.
- If you choose percent for this parameter, and `50` for the *x-position* parameter, the click will always be in the middle of the x-axis.

Name	Data Type	Values	Default
y-position	Integer	–	50

- With this parameter, you can specify where in the component you should click.
- Use this parameter to enter a value for the y-axis (the vertical position).
- Using the y-units parameter, you can specify whether the value should be in pixels or in percent.
- For example, if you enter `50` and then `percent` (without quotes), then the click will be performed in the middle of the y-axis.
- All positions are calculated from the upper left of the component, with the positive y-axis stretching toward the bottom of the component.

We recommend not using `0` (pixels and percent) or `100`(percent) positions.

Name	Data Type	Values	Default
y-units	String	percent pixel	percent

- Use this parameter to specify whether you want your values for the y-coordinate to be in pixels or percent.
- If you choose percent for this parameter, and `50` for the *y-position* parameter, the click will always be in the middle of the y-axis.

2.1.2.19 Drag

`(Graphics Component)`

- This action drags the component you specify.
- The cursor is moved over the component.
- The mouse button you specify is held, and the component is dragged.
- You **must** follow this action with a *drop* action

This action is unsupported for the HTML toolkit

Parameters

Name	Data Type	Values	Default
Mouse Button	Integer	1 2 3	1

- Use this parameter to select which mouse button you want to click with.
  - `1` = left mouse button
  - `2` = middle mouse button
  - `3` = right mouse button
- The mouse button you use to click with can also have an effect on what happens.
- The button to click will depend on your AUT and what the click should achieve.

Do not enter quotes around the mouse button numbers, enter them in plain text.

Name	Data Type	Values	Default
Modifier Keys	String	none shift control alt meta cmd mod	none

- Use this parameter to specify a key or keys to be pressed at the same time as the action is being executed.
- You can use this parameter to specify the following keys, or combinations thereof:

  `shift`

  `control` This is the first modifier for Windows and Linux

  `cmd` This is the first modifier for Mac

  `mod` This is the platform-independent first modifier. It corresponds to »CONTROL« for Windows and Linux, and »CMD« for Mac.

  `meta`

  `alt`
- You can also select "*none*" if you do not need a modifier.
- You can enter multiple modifiers for a key combination by entering a space-separated list of modifiers, e.g. `alt shift`.

Do not use quotes around the modifiers, enter them in plain text

Name	Data Type	Values	Default
x-position	Integer	–	50

- With this parameter, you can specify where in the component you should click.
- Use this parameter to enter a value for the x-axis (the horizontal position).
- Using the x-units parameter, you can specify whether the value should be in pixels or in percent.
- For example, if you enter `50` and then `percent` (without quotes) , then the click will be performed in the middle of the x-axis.
- All positions are calculated from the upper left of the component, with the positive x-axis stretching toward the right side of the component.

We recommend not using `0` (pixels and percent) or `100`(percent) positions.

Name	Data Type	Values	Default
x-units	String	percent pixel	percent

- Use this parameter to specify whether you want your values for the x-coordinate to be in pixels or percent.
- If you choose percent for this parameter, and `50` for the *x-position* parameter, the click will always be in the middle of the x-axis.

Name	Data Type	Values	Default
y-position	Integer	–	50

- With this parameter, you can specify where in the component you should click.
- Use this parameter to enter a value for the y-axis (the vertical position).
- Using the y-units parameter, you can specify whether the value should be in pixels or in percent.
- For example, if you enter `50` and then `percent` (without quotes), then the click will be performed in the middle of the y-axis.
- All positions are calculated from the upper left of the component, with the positive y-axis stretching toward the bottom of the component.

We recommend not using `0` (pixels and percent) or `100`(percent) positions.

Name	Data Type	Values	Default
y-units	String	percent pixel	percent

- Use this parameter to specify whether you want your values for the y-coordinate to be in pixels or percent.
- If you choose percent for this parameter, and `50` for the *y-position* parameter, the click will always be in the middle of the y-axis.

2.1.2.20 Drop

`(Graphics Component)`

- This action drops a dragged item onto the component you specify.
- The cursor is moved over the component.
- The mouse button is let loose, and the component is dropped.
- Precede this action with a *drag* action

This action is unsupported for the HTML toolkit

Parameters

Name	Data Type	Values	Default
x-position	Integer	–	50

- With this parameter, you can specify where in the component you should click.
- Use this parameter to enter a value for the x-axis (the horizontal position).
- Using the x-units parameter, you can specify whether the value should be in pixels or in percent.
- For example, if you enter `50` and then `percent` (without quotes) , then the click will be performed in the middle of the x-axis.
- All positions are calculated from the upper left of the component, with the positive x-axis stretching toward the right side of the component.

We recommend not using `0` (pixels and percent) or `100`(percent) as positions.

Name	Data Type	Values	Default
x-units	String	percent pixel	percent

- Use this parameter to specify whether you want your values for the x-coordinate to be in pixels or percent.
- If you choose percent for this parameter, and `50` for the *x-position* parameter, the click will always be in the middle of the x-axis.

Name	Data Type	Values	Default
y-position	Integer	–	50

- With this parameter, you can specify where in the component you should click.
- Use this parameter to enter a value for the y-axis (the vertical position).
- Using the y-units parameter, you can specify whether the value should be in pixels or in percent.
- For example, if you enter `50` and then `percent` (without quotes), then the click will be performed in the middle of the y-axis.
- All positions are calculated from the upper left of the component, with the positive y-axis stretching toward the bottom of the component.

We recommend not using `0` (pixels and percent) or `100`(percent) positions.

Name	Data Type	Values	Default
y-units	String	percent pixel	percent

- Use this parameter to specify whether you want your values for the y-coordinate to be in pixels or percent.
- If you choose percent for this parameter, and `50` for the *y-position* parameter, the click will always be in the middle of the y-axis.

Name	Data Type	Values	Default
Delay before drop (milliseconds)	Integer	–	100

- Use this parameter to wait before dropping the item you have dragged.
- This can be useful to give the user interface time to scroll etc.
- Give the time to delay in milliseconds.

2.1.2.21 Select Context Menu Entry by Indexpath

`(Graphics Component)`

- Use this action to open the context-sensitive menu from the currently active component and to select an item from it.
- The menu item is selected using its indexpath.
- This is realized by:
  - Right-clicking in the component.
  - Navigating to the menu item specified.
  - Left-clicking on the menu option.
- If you want to select an item from the context-menu on a specific node or cell, then you must first specify that the node or cell you want be selected using the appropriate select action on the *tree* or *table* component.
- Depending on your AUT, you may have to enter `0` as the click count for the select action on the *tree* or *table* component.

This action is unsupported for the HTML toolkit

Parameters

Name	Data Type	Values	Default
Indexpath	String	–	none

Use this parameter to enter the path to the context menu item you want to select. Make sure you give the whole menupath (start from the very beginning of the menu).

- Enter the path to the item as an indexpath.
- Use slash `'/'` as a path separator (to separate parent nodes from child nodes).
- For example, `1/2` (without quotes).

The first node is '1' (without quotes)

Name	Data Type	Values	Default
Mouse Button	Integer	1 2 3	3

- Use this parameter to select which mouse button you want to click with.
  - `1` = left mouse button
  - `2` = middle mouse button
  - `3` = right mouse button
- The mouse button you use to click with can also have an effect on what happens.
- The button to click will depend on your AUT and what the click should achieve.

Do not enter quotes around the mouse button numbers, en them in plain text.

2.1.2.22 Select Context Menu Entry by Indexpath (Specify Position)

`(Graphics Component)`

- Use this action to open the context-sensitive menu from the currently active component and to select an item from it.
- The menu item is selected using its indexpath.
- This is realized by:
  - Clicking in the component with the mouse button you specify.
  - The position to click is specified by the x- and y- coordinate parameters.
  - Navigating to the menu item specified.
  - Left-clicking on the menu option.
- This action will open the context menu at the position you specify in the component.
- For some components (e.g. trees), opening a context menu somewhere else in the component can have a different effect than when you open the context menu on a specific node.

- If you want to open or check a context menu on a selected node or cell in a component, then use the action which selects the context menu without specifying the coordinate position.

This action is unsupported for the HTML toolkit

Parameters

Name	Data Type	Values	Default
x-position	Integer	–	50

- With this parameter, you can specify where in the component you should click.
- Use this parameter to enter a value for the x-axis (the horizontal position).
- Using the x-units parameter, you can specify whether the value should be in pixels or in percent.
- For example, if you enter `50` and then `percent` (without quotes) , then the click will be performed in the middle of the x-axis.
- All positions are calculated from the upper left of the component, with the positive x-axis stretching toward the right side of the component.

We recommend not using `0` (pixels and percent) or `100`(percent) as positions.

Name	Data Type	Values	Default
x-units	String	percent pixel	percent

- Use this parameter to specify whether you want your values for the x-coordinate to be in pixels or percent.
- If you choose percent for this parameter, and `50` for the *x-position* parameter, the click will always be in the middle of the x-axis.

Name	Data Type	Values	Default
y-position	Integer	–	50

- With this parameter, you can specify where in the component you should click.
- Use this parameter to enter a value for the y-axis (the vertical position).

- Using the y-units parameter, you can specify whether the value should be in pixels or in percent.
- For example, if you enter `50` and then `percent` (without quotes), then the click will be performed in the middle of the y-axis.
- All positions are calculated from the upper left of the component, with the positive y-axis stretching toward the bottom of the component.

We recommend not using `0` (pixels and percent) or `100`(percent) positions.

Name	Data Type	Values	Default
y-units	String	percent pixel	percent

- Use this parameter to specify whether you want your values for the y-coordinate to be in pixels or percent.
- If you choose percent for this parameter, and `50` for the *y-position* parameter, the click will always be in the middle of the y-axis.

Name	Data Type	Values	Default
Indexpath	String	–	none

- Enter the path to the item as an indexpath.
- Use slash `'/'` as a path separator (to separate parent nodes from child nodes).
- For example, `1/2` (without quotes).

The first node is ’1’ (without quotes)

Name	Data Type	Values	Default
Mouse Button	Integer	1 2 3	3

- Use this parameter to select which mouse button you want to click with.
  - `1` = left mouse button
  - `2` = middle mouse button
  - `3` = right mouse button
- The mouse button you use to click with can also have an effect on what happens.

- The button to click will depend on your AUT and what the click should achieve.

Do not enter quotes around the mouse button numbers, enter them in plain text.

2.1.2.23 Select Context Menu Entry by Textpath

`(Graphics Component)`

- Use this action to open the context-sensitive menu from the currently active component and to select an item from it.
- The menu item is selected using its textpath.
- This is realized by:
  - Right-clicking in the component, at the place where the cursor currently is.
  - Navigating to the menu item specified.
  - Left-clicking on the menu option.
- If you want to select an item from the context-menu on a specific node or cell, then you must first specify that the node or cell you want be selected using the appropriate select action on the *tree* or *table* component.
- Depending on your AUT, you may have to enter `0` as the click count for the select action on the *tree* or *table* component.

This action is unsupported for the HTML toolkit

Parameters

Name	Data Type	Values	Default
Textpath	String	–	none

Use this parameter to enter the path to the context menu item you want to select. Make sure you enter the whole path.

- Enter the path to the item as a textpath.

- Use slash `'/'` as a path separator (to separate parent nodes from child nodes).
- For example, `File/Open` or `Category/Horror` (without quotes).
- Either make sure that your path is written exactly as it appears in the interface, or use a regular expression to match the text.
- Each segment of the path will be used to find a corresponding node, using the operator provided.

Name	Data Type	Values	Default
Operator	String	equals not equals matches simple match	equals

- The *operator* parameter has four possible values.
- "*not equals*" looks for something that does *not exactly* match.
- "*equals*" looks for an *exact* match.
- Select "*simple match*" to use a simple match expression (→ page **??**) .
- Select "*matches*" to use a regular expression (→ page **??**) .
- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

Name	Data Type	Values	Default
Mouse Button	Integer	1 2 3	3

- Use this parameter to select which mouse button you want to click with.
  - `1` = left mouse button
  - `2` = middle mouse button
  - `3` = right mouse button
- The mouse button you use to click with can also have an effect on what happens.
- The button to click will depend on your AUT and what the click should achieve.

Do not enter quotes around the mouse button numbers, en them in plain text.

2.1.2.24 Select Context Menu Entry by Textpath (Specify Position)

`(Graphics Component)`

- Use this action to open the context-sensitive menu from the currently active component and to select an item from it.
- The menu item is selected using its textpath.
- This is realized by:
  - Clicking in the component with the mouse button you specify.
  - The position to click is specified by the x- and y- coordinate parameters.
  - Navigating to the menu item specified.
  - Left-clicking on the menu option.
- This action will open the context menu at the position you specify in the component.
- For some components (e.g. trees), opening a context menu somewhere else in the component can have a different effect than when you open the context menu on a specific node.
- If you want to open or check a context menu on a selected node or cell in a component, then use the action which selects the context menu without specifying the coordinate position.

This action is unsupported for the HTML toolkit

Parameters

Name	Data Type	Values	Default
x-position	Integer	–	50

- With this parameter, you can specify where in the component you should click.
- Use this parameter to enter a value for the x-axis (the horizontal position).
- Using the x-units parameter, you can specify whether the value should be in pixels or in percent.

- For example, if you enter `50` and then `percent` (without quotes) , then the click will be performed in the middle of the x-axis.
- All positions are calculated from the upper left of the component, with the positive x-axis stretching toward the right side of the component.

We recommend not using `0` (pixels and percent) or `100`(percent) positions.

Name	Data Type	Values	Default
x-units	String	percent pixel	percent

- Use this parameter to specify whether you want your values for the x-coordinate to be in pixels or percent.
- If you choose percent for this parameter, and `50` for the *x-position* parameter, the click will always be in the middle of the x-axis.

Name	Data Type	Values	Default
y-position	Integer	–	50

- With this parameter, you can specify where in the component you should click.
- Use this parameter to enter a value for the y-axis (the vertical position).
- Using the y-units parameter, you can specify whether the value should be in pixels or in percent.
- For example, if you enter `50` and then `percent` (without quotes), then the click will be performed in the middle of the y-axis.
- All positions are calculated from the upper left of the component, with the positive y-axis stretching toward the bottom of the component.

We recommend not using `0` (pixels and percent) or `100`(percent) positions.

Name	Data Type	Values	Default
y-units	String	percent pixel	percent

- Use this parameter to specify whether you want your values for the y-coordinate to be in pixels or percent.

- If you choose percent for this parameter, and `50` for the *y-position* parameter, the click will always be in the middle of the y-axis.

Name	Data Type	Values	Default
Menupath	String	–	none

Use this parameter to enter the path to the context menu item you want to select. Make sure you enter the whole path.

- Enter the path to the item as a textpath.
- Use slash `'/'` as a path separator (to separate parent nodes from child nodes).
- For example, `File/Open` or `Category/Horror` (without quotes).
- Either make sure that your path is written exactly as it appears in the interface, or use a regular expression to match the text.
- Each segment of the path will be used to find a corresponding node, using the operator provided.

Name	Data Type	Values	Default
Operator	String	equals not equals matches simple match	equals

- The *operator* parameter has four possible values.
- "*not equals*" looks for something that does *not exactly* match.
- "*equals*" looks for an *exact* match.
- Select "*simple match*" to use a simple match expression (→ page **??**) .
- Select "*matches*" to use a regular expression (→ page **??**) .
- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

Name	Data Type	Values	Default
Mouse Button	Integer	1 2 3	3

- Use this parameter to select which mouse button you want to click with.
  - `1` = left mouse button
  - `2` = middle mouse button
  - `3` = right mouse button
- The mouse button you use to click with can also have an effect on what happens.

- The button to click will depend on your AUT and what the click should achieve.

Do not enter quotes around the mouse button numbers, en them in plain text.

2.1.2.25 Show Text

(Graphics Component)

- Use this action to produce an area with text you define.
- This is mostly useful for demonstration purposes.

Parameters

Name	Data Type	Values	Default
Text	String	–	none

- Use this parameter to enter the text you want to show.

Name	Data Type	Values	Default
Text Size (in points)	Integer	–	14

- Use this parameter to define how large the text should be.
- Give the font size you want to use in points.

Name	Data Type	Values	Default
Time per Word (in milliseconds)	Integer	–	500

- Use this parameter to define how long the text should be shown for.
- The time is calculated per word.
- Enter the value (in milliseconds) that you want to allow per word.
- For 3 seconds, enter 300.
- The time per word is multiplied by the amount of words in your text to give the total amount of time the text is shown for.

Name	Data Type	Values	Default
Window Width (in pixels)	Integer	–	250

- Use this parameter to define how wide the text window should be.
- The width is given in pixels.

2.1.2.26 Store Property

`(Graphics Component)`

- Use this action to store the value of a specific property of a component.
- You can then use this value later in your test, or check whether two values you store are the same or different using the actions to Check Numeric Values and Check String Values on the Application component.
- Every component has properties (attributes). These can include the text size, the size (width/height) of the component, the color etc.
- You must first find out the name of the property you want to check, and then what value it should have. You can find out what properties and values a component has in the Object Mapping Mode (for Swing and SWT/RCP AUT's). When you collect a component, its property information is displayed in the Properties View.

If the property is *private*, it cannot be stored, the remote control component has no access to it.

Parameters

Name	Data Type	Values	Default
Variable Name	Variable	–	none

- Enter the name you want to give to this variable.
- Variable names may only contain letters, numbers and underscores.
- You can then enter this variable name as data for other Test Steps.
- When you enter the variable name as data, place a dollar sign before it.
- The data associated with this variable name remains the same until the Project is closed, or until you overwrite the name by using it for another value.

Name	Data Type	Values	Default
Property Name	String	–	none

- Enter the name of the property whose value you want to store.

2.1.2.27 Wait for Component

`(Graphics Component)`

- The wait for component action lets you choose a component which you need for the next steps.
- The test waits until the component is present/loaded before continuing with the next Test Step.

Parameters

Name	Data Type	Values	Default
Timeout in ms	Integer	–	1000

- Enter the amount of time (in milliseconds) to wait before the Test Step fails.
- If the component does not appear in this time frame, the Test Step is unsuccessful.

Name	Data Type	Values	Default
Delay after Visibility	Integer	–	200

- Sometimes a widget is not ready to receive events (mouse clicks, key presses) until a few moments after becoming visible.
- Use this parameter to set the amount of time (in milliseconds) to wait once the widget has appeared before continuing.

Used By

(HTML) Anchor (html)
(SWT) Tool Item (swt)
(SWT) Tree (swt)
(SWT) Tree Table (swt)
Button Component (concrete)
Combo Component (concrete)
Figure Canvas (gef)
List Component (concrete)
Tab Component (concrete)
Table Component (concrete)
Text Component (concrete)
Text Input Component (concrete)
Tree Component (concrete)

2.1.3 Text Component

Description:

- This is an abstract component.
- It is implemented by all components which contain a text value that can be verified.

Synopsis:

- Text Component (abstract)
  - Graphics Component (abstract)

New Actions

Name	Type	Parameters
Check Text (→ page 76)	c	String:Text String:Operator
Store Value (→ page 77)	e	Variable:Variable Name

Inherited Actions

Name	Type	Parameters	Inherited from
Check Enablement (→ page 23)	c	Boolean:Enabled	Graphics Component (abstract)
Check Enablement of Context Menu Entry by Indexpath (→ page 24)	c	String:Indexpath Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)
Check Enablement of Context Menu Entry by Indexpath (Specify Position) (→ page 25)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Indexpath Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)

Check Enablement of Context Menu Entry by Textpath (→ page 28)	c	String:Textpath String:Operator Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)
Check Enablement of Context Menu Entry by Textpath (Specify Position) (→ page 29)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Textpath String:Operator Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)
Check Existence (→ page 32)	c	Boolean:Exists	Graphics Component (abstract)
Check Existence of Context Menu Entry by Indexpath (→ page 33)	c	String:Indexpath Boolean:Exists Integer:Mouse Button	Graphics Component (abstract)
Check Existence of Context Menu Entry by Indexpath (Specify Position) (→ page 34)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Indexpath Boolean:Exists Integer:Mouse Button	Graphics Component (abstract)
Check Existence of Context Menu Entry by Textpath (→ page 37)	c	String:Textpath String:Operator Boolean:Exists Integer:Mouse Button	Graphics Component (abstract)
Check Existence of Context Menu Entry by Textpath (Specify Position) (→ page 38)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Textpath String:Operator Boolean:Exists Integer:Mouse Button	Graphics Component (abstract)

Check Focus (→ page 41)	c	Boolean:Has Focus	Graphics Component (abstract)
Check Property (→ page 42)	c	String:Property Name String:Property Value String:Operator	Graphics Component (abstract)
Check Selection of Context Menu Entry by Indexpath (→ page 43)	c	String:Indexpath Boolean:Selected Integer:Mouse Button	Graphics Component (abstract)
Check Selection of Context Menu Entry by Indexpath (Specify Position) (→ page 45)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Indexpath Boolean:Selected Integer:Mouse Button	Graphics Component (abstract)
Check Selection of Context Menu Entry by Textpath (→ page 47)	c	String:Textpath String:Operator Boolean:Selected Integer:Mouse Button	Graphics Component (abstract)
Check Selection of Context Menu Entry by Textpath (Specify Position) (→ page 49)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Textpath String:Operator Boolean:Selected Integer:Mouse Button	Graphics Component (abstract)

Click (→ page 52)	e	Integer:Number of Clicks Integer:Mouse Button	Graphics Component (abstract)
Click in Component (→ page 53)	e	Integer:Number of Clicks Integer:Mouse Button Integer:x-position String:x-units Integer:y-position String:y-units	Graphics Component (abstract)
Drag (→ page 55)	e	Integer:Mouse Button String:Modifier Keys Integer:x-position String:x-units Integer:y-position String:y-units	Graphics Component (abstract)
Drop (→ page 58)	e	Integer:x-position String:x-units Integer:y-position String:y-units Integer:Delay before drop (milliseconds)	Graphics Component (abstract)
Select Context Menu Entry by Indexpath (→ page 60)	e	String:Indexpath Integer:Mouse Button	Graphics Component (abstract)
Select Context Menu Entry by Indexpath (Specify Position) (→ page 61)	e	Integer:x-position String:x-units Integer:y-position String:y-units String:Indexpath Integer:Mouse Button	Graphics Component (abstract)

Select Context Menu Entry by Textpath (→ page 64)	e	String:Textpath String:Operator Integer:Mouse Button	Graphics Component (abstract)
Select Context Menu Entry by Textpath (Specify Position) (→ page 66)	e	Integer:x-position String:x-units Integer:y-position String:y-units String:Menupath String:Operator Integer:Mouse Button	Graphics Component (abstract)
Show Text (→ page 69)	e	String:Text Integer:Text Size (in points) Integer:Time per Word (in milliseconds) Integer:Window Width (in pixels)	Graphics Component (abstract)
Store Property (→ page 70)	e	Variable:Variable Name String:Property Name	Graphics Component (abstract)
Wait for Component (→ page 71)	e	Integer:Timeout in ms Integer:Delay after Visibility	Graphics Component (abstract)

2.1.3.1 Check Text

`(Text Component)`

- Use this action to check whether the text in a component matches a given value.

If you use this action on a table, then the currently selected cell will be checked

Parameters

Name	Data Type	Values	Default
Text	String	–	none

- Enter the text you want to check against the text in the component.
- This parameter will be compared to the text in the component, using the Operator provided.

Name	Data Type	Values	Default
Operator	String	equals not equals matches simple match	equals

- The *operator* parameter has four possible values.
- "*not equals*" looks for something that does *not exactly* match.
- "*equals*" looks for an *exact* match.
- Select "*simple match*" to use a simple match expression (→ page **??**) .
- Select "*matches*" to use a regular expression (→ page **??**) .
- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

2.1.3.2 Store Value

`(Text Component)`

- Use this action to read the value out of a field so that you can use it as data for other Test Steps.

Parameters

Name	Data Type	Values	Default
Variable Name	Variable	–	none

- Enter the name you want to give to this variable.
- Variable names may only contain letters, numbers and underscores.
- You can then enter this variable name as data for other Test Steps.
- When you enter the variable name as data, place a dollar sign before it.

- The data associated with this variable name remains the same until the Project is closed, or until you overwrite the name by using it for another value.

Used By

(HTML) Anchor (html)
(SWT) Tool Item (swt)
Button Component (concrete)
Combo Component (concrete)
List Component (concrete)
Table Component (concrete)
Text Component (concrete)
Text Input Component (concrete)

2.1.4 Text Input Component

Description:

- This is an abstract component.
- It is implemented by all components whose properties or values can be altered.

Synopsis:

- Text Input Component (abstract)
  - Text Component (abstract)
    - ⋆ Graphics Component (abstract)

New Actions

Name	Type	Parameters
Check Editability (→ page 84)	c	Boolean:Editable
Input Text (→ page 84)	e	String:Text
Replace Text (→ page 85)	e	String:Text

Inherited Actions

Name	Type	Parameters	Inherited from
Check Enablement (→ page 23)	c	Boolean:Enabled	Graphics Component (abstract)
Check Enablement of Context Menu Entry by Indexpath (→ page 24)	c	String:Indexpath Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)
Check Enablement of Context Menu Entry by Indexpath (Specify Position) (→ page 25)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Indexpath Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)

Check Enablement of Context Menu Entry by Textpath (→ page 28)	c	String:Textpath String:Operator Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)
Check Enablement of Context Menu Entry by Textpath (Specify Position) (→ page 29)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Textpath String:Operator Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)
Check Existence (→ page 32)	c	Boolean:Exists	Graphics Component (abstract)
Check Existence of Context Menu Entry by Indexpath (→ page 33)	c	String:Indexpath Boolean:Exists Integer:Mouse Button	Graphics Component (abstract)
Check Existence of Context Menu Entry by Indexpath (Specify Position) (→ page 34)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Indexpath Boolean:Exists Integer:Mouse Button	Graphics Component (abstract)
Check Existence of Context Menu Entry by Textpath (→ page 37)	c	String:Textpath String:Operator Boolean:Exists Integer:Mouse Button	Graphics Component (abstract)
Check Existence of Context Menu Entry by Textpath (Specify Position) (→ page 38)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Textpath String:Operator Boolean:Exists Integer:Mouse Button	Graphics Component (abstract)

Check Focus (→ page 41)	c	Boolean:Has Focus	Graphics Component (abstract)
Check Property (→ page 42)	c	String:Property Name String:Property Value String:Operator	Graphics Component (abstract)
Check Selection of Context Menu Entry by Indexpath (→ page 43)	c	String:Indexpath Boolean:Selected Integer:Mouse Button	Graphics Component (abstract)
Check Selection of Context Menu Entry by Indexpath (Specify Position) (→ page 45)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Indexpath Boolean:Selected Integer:Mouse Button	Graphics Component (abstract)
Check Selection of Context Menu Entry by Textpath (→ page 47)	c	String:Textpath String:Operator Boolean:Selected Integer:Mouse Button	Graphics Component (abstract)
Check Selection of Context Menu Entry by Textpath (Specify Position) (→ page 49)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Textpath String:Operator Boolean:Selected Integer:Mouse Button	Graphics Component (abstract)

Check Text (→ page 76)	c	String:Text String:Operator	Text Component (abstract)
Click (→ page 52)	e	Integer:Number of Clicks Integer:Mouse Button	Graphics Component (abstract)
Click in Component (→ page 53)	e	Integer:Number of Clicks Integer:Mouse Button Integer:x-position String:x-units Integer:y-position String:y-units	Graphics Component (abstract)
Drag (→ page 55)	e	Integer:Mouse Button String:Modifier Keys Integer:x-position String:x-units Integer:y-position String:y-units	Graphics Component (abstract)
Drop (→ page 58)	e	Integer:x-position String:x-units Integer:y-position String:y-units Integer:Delay before drop (milliseconds)	Graphics Component (abstract)

Select Context Menu Entry by Indexpath (→ page 60)	e	String:Indexpath Integer:Mouse Button	Graphics Component (abstract)
Select Context Menu Entry by Indexpath (Specify Position) (→ page 61)	e	Integer:x-position String:x-units Integer:y-position String:y-units String:Indexpath Integer:Mouse Button	Graphics Component (abstract)
Select Context Menu Entry by Textpath (→ page 64)	e	String:Textpath String:Operator Integer:Mouse Button	Graphics Component (abstract)
Select Context Menu Entry by Textpath (Specify Position) (→ page 66)	e	Integer:x-position String:x-units Integer:y-position String:y-units String:Menupath String:Operator Integer:Mouse Button	Graphics Component (abstract)
Show Text (→ page 69)	e	String:Text Integer:Text Size (in points) Integer:Time per Word (in milliseconds) Integer:Window Width (in pixels)	Graphics Component (abstract)
Store Property (→ page 70)	e	Variable:Variable Name String:Property Name	Graphics Component (abstract)
Store Value (→ page 77)	e	Variable:Variable Name	Text Component (abstract)

Wait for Component (→ page 71)	e	Integer:Timeout in ms Integer:Delay after Visibility	Graphics Component (abstract)

2.1.4.1 Check Editability

```
(Text Input Component)
```

- Use this action to check whether or not the component is editable

Parameters

Name	Data Type	Values	Default
Editable	Boolean	true false	true

- Set the parameter to true if you expect the component to be editable.
- Set the parameter to false if you expect the component *not* to be editable.

2.1.4.2 Input Text

```
(Text Input Component)
```

- Use this action to enter text into a component.
- The text you want to enter is given as a parameter.
- This is realized by:
  - Clicking once on the component if it is not already ready to support keyboard input.
  - Entering the text at the current cursor position.
- To delete any text already in the component, use the action”*replace text*”.
- To enter text before/after previous text, see ”*Insert Text Before/After Pattern*” in the ”*Text Field*” component.

This action is unsupported for combo-boxes in the HTML toolkit

Parameters

Name	Data Type	Values	Default
Text	String	–	none

- Use this parameter to specify the text you want to enter into the component.

2.1.4.3 Replace Text

`(Text Input Component)`

- Use this action to replace any text already in the text field.
- The text you want to enter is given as a parameter.
- This is realized by:
  - Selecting any text already in the component.
  - Entering the text you specified. This effectively overwrites the previous text in the component.
- This means that any previous text in the component is deleted.
- To enter text before/after previous text, see "*Insert Text Before/After Pattern*" in the "*Text Field*" component.

This action is unsupported for combo-boxes in the HTML toolkit

If you use this action on a table, then the text will be entered into the currently selected cell.

Parameters

Name	Data Type	Values	Default
Text	String	–	none

- Use this parameter to specify the text you want to enter into the component.
- Any previous text will be deleted.

Used By

Combo Component (concrete)
Table Component (concrete)
Text Input Component (concrete)

2.2 concrete Toolkit

The concrete toolkit contains components and actions which are available in all other supported toolkits. An action from the concrete toolkit can therefore be used in a test for Swing, SWT and HTML applications.

2.2.1 Application

Description:

- The *Application* component is the container for all other components.
- It represents the AUT as a whole.

Synopsis:

- Application (concrete)

New Actions

Name	Type	Parameters
Activate (→ page 89)	e	String:Activation Method
Check Existence of Window (→ page 90)	e	String:Title String:Operator Boolean:Exists
Check Numeric Values (→ page 91)	e	String:Value 1 String:Comparison Method String:Value 2
Check String Values (→ page 92)	e	String:Value 1 String:Value 2 Operator String:Value 2
Click in Active Window (→ page 93)	e	Integer:Number of Clicks Integer:Mouse Button Integer:x-position String:x-units Integer:y-position String:y-units
Copy Text to Clipboard (→ page 95)	e	String:Text
Execute External Command (→ page 96)	e	String:Command Integer:Expected Exit Code Boolean:Local Integer:Timeout in ms
External Input Text (→ page 97)	e	String:Text
External Key Combination (→ page 98)	e	String:Modifier Keys String:Base Key

Input Text (→ page 100)	e	String:Text
Key Combination (→ page 101)	e	String:Modifier Keys String:Base Key
Manual Test Step (→ page 103)	e	String:Action to perform String:Expected Behavior Integer:Timeout in ms
Pause Test Execution (→ page 103)	e	
Prepare for AUT termination (→ page 104)	e	
Read Timer (→ page 105)	e	String:Timer Name Variable:Timer Value Variable Name
Restart (→ page 106)	e	
Set Toggle Key (→ page 106)	e	Integer:Key Boolean:Activate Key
Start Timer (→ page 107)	e	String:Timer Name Variable:Start Time Variable Name
Store Value (→ page 108)	e	Variable:Variable Name String:Value
Synchronize termination and re-start of AUT (→ page 109)	e	Integer:Timeout in ms
Take Screenshot (→ page 110)	e	String:Destination Integer:Delay String:File Access Integer:Scaling Factor Boolean:Create Directories
Take Screenshot of Active Window (→ page 112)	e	String:Destination Integer:Delay String:File Access Integer:Scaling Factor Boolean:Create Directories Integer:Margin Top Integer:Margin Right Integer:Margin Bottom Integer:Margin Left

Wait (→ page 114)	e	Integer:Millisecs
Wait for Window (→ page 114)	e	String:Title String:Operator Integer:Timeout in ms Integer:Delay after Visibility
Wait for Window Activation (→ page 116)	e	String:Title String:Operator Integer:Timeout in ms Integer:Delay after Visibility
Wait for Window to Close (→ page 117)	e	String:Title String:Operator Integer:Timeout in ms Integer:Delay after Closure

Inherited Actions

none

2.2.1.1 Activate

`(Application)`

- When the ITE and AUT Agent are running on the same computer, you may need to bring the AUT into focus before a test can begin.
- Exactly how to *activate* the application can vary from one system to another.
- This action offers various different ways of activating the application.
- This action is performed by clicking in the location indicated in the *activation method* parameter.
- In the AUT configuration (→*User Manual* p. 50), the default is that no activation is carried out.
- You can set an activation method for the whole AUT in its configuration, and use this in your Test Steps (e.g. set the activation method parameter to *"AUT_DEFAULT"*) or you can use a different activation method.
- In this way, you can centrally define a default for a given operating system, which you can easily change when you test on another operating system.

- Using the activate action, you can also specify activation methods which differ from the default.

This action is used to ensure that an AUT has the current focus. It cannot be used to maximize a minimized AUT.

Parameters

Name	Data Type	Values	Default
Activation Method	String	AUT_DEFAULT NONE TITLEBAR NW NE SW SE CENTER	AUT_DEFAULT

- Use this parameter to specify how you want the AUT to be activated.
- You have a choice of possible values:

Value	Description
`NONE`	does not activate the AUT: in effect, this parameter does nothing.
`AUT_DEFAULT`	uses the default from the AUT configuration
`NE`	click in the northeast corner of the AUT window
`NW`	click in the northwest corner of the AUT window
`SE`	click in the southeast corner
`SW`	click in the southwest corner
`TITLEBAR`	click in the window's titlebar. This does not work with small or non-existent titlebars.

2.2.1.2 Check Existence of Window

`(Application)`

- Use this action to check the existence of a window based on its title.

Parameters

Name	Data Type	Values	Default
Title	String	–	none

- Enter the title of the window you are waiting for.
- The title you provide will be compared to the title bar of each window as it opens, using the Operator provided.

Name	Data Type	Values	Default
Operator	String	equals not equals matches simple match	equals

- The *operator* parameter has four possible values.
- "*not equals*" looks for something that does *not exactly* match.
- "*equals*" looks for an *exact* match.
- Select "*simple match*" to use a simple match expression (→ page **??**) .
- Select "*matches*" to use a regular expression (→ page **??**) .
- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

Name	Data Type	Values	Default
Exists	Boolean	true false	true

- Set this parameter to true if you expect the window to exist.
- Set this parameter to false if you do *not* expect the window to exist.

2.2.1.3 Check Numeric Values

`(Application)`

- Use this action to compare two numeric values during your test.
- This action can be used with the *start timer* and *read timer* actions to compare the time difference to a value you specify (i.e. to measure the performance of your application).
- You can also compare other variables you have read from your application with this action.

Parameters

Name	Data Type	Values	Default
Value 1	String	–	none

- Enter the first value or variable you want this action to check.

Name	Data Type	Values	Default
Comparison Method	String	less than less or equal than equal to greater or equal than greater than	equal to

- Select the comparison method you want to use to check the two values or variables.

Name	Data Type	Values	Default
Value 2	String	–	none

- Enter the second value or variable you want this action to check.

2.2.1.4 Check String Values

`(Application)`

Use this action to compare the values of two strings in your test. This can be used if you want to check the value of a variable, for example.

Parameters

Name	Data Type	Values	Default
Value 1	String	–	none

- Enter the value or variable you want to perform the check against.
- With the other parameters, you can enter how the value should be compared and what it should be compared to.

Name	Data Type	Values	Default
Value 2 Operator	String	equals not equals matches simple match	equals

Use this parameter to define the operator for the *value 2* parameter.

- The *operator* parameter has four possible values.
- *"not equals"* looks for something that does *not exactly* match.
- *"equals"* looks for an *exact* match.
- Select *"simple match"* to use a simple match expression (→ page **??**) .
- Select *"matches"* to use a regular expression (→ page **??**) .
- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

Name	Data Type	Values	Default
Value 2	String	–	none

- Enter the value or variable you want to use to perform the check against the *value 1* parameter.
- If you want to check that value one begins with `A` and ends with `Z` then you could use *matches* as the operator parameter, and enter `A.*Z` as the parameter for *value 2*.

2.2.1.5 Click in Active Window

`(Application)`

- Use this action to send one or more clicks with any mouse button to the currently active window.
- You can specify where to click, how many times, and with which mouse button.

Parameters

Name	Data Type	Values	Default
Number of Clicks	Integer	–	1

- Enter the amount of clicks that should be performed on the component or item.
- If you enter `0`, the mouse pointer will just be moved over the specified component or item.
- Entering `1` (without quotes) will single-click on the component or item.
- Entering `2` (without quotes) will double-click on the component or item.

Name	Data Type	Values	Default
Mouse Button	Integer	1 2 3	1

- Use this parameter to select which mouse button you want to click with.
  - `1` = left mouse button
  - `2` = middle mouse button
  - `3` = right mouse button
- The mouse button you use to click with can also have an effect on what happens.
- The button to click will depend on your AUT and what the click should achieve.

Do not enter quotes around the mouse button numbers, enter them in plain text.

Name	Data Type	Values	Default
x-position	Integer	–	50

- With this parameter, you can specify where in the component you should click.
- Use this parameter to enter a value for the x-axis (the horizontal position).
- Using the x-units parameter, you can specify whether the value should be in pixels or in percent.
- For example, if you enter `50` and then `percent` (without quotes) , then the click will be performed in the middle of the x-axis.
- All positions are calculated from the upper left of the component, with the positive x-axis stretching toward the right side of the component.

We recommend not using `0` (pixels and percent) or `100`(percent) as positions.

Name	Data Type	Values	Default
x-units	String	percent pixel	percent

- Use this parameter to specify whether you want your values for the x-coordinate to be in pixels or percent.
- If you choose percent for this parameter, and `50` for the *x-position* parameter, the click will always be in the middle of the x-axis.

Name	Data Type	Values	Default
y-position	Integer	–	50

- With this parameter, you can specify where in the component you should click.
- Use this parameter to enter a value for the y-axis (the vertical position).
- Using the y-units parameter, you can specify whether the value should be in pixels or in percent.
- For example, if you enter `50` and then `percent` (without quotes), then the click will be performed in the middle of the y-axis.
- All positions are calculated from the upper left of the component, with the positive y-axis stretching toward the bottom of the component.

We recommend not using `0` (pixels and percent) or `100`(percent) positions.

Do not use this action with the value 100% on Linux systems, as this causes a timeout.

Name	Data Type	Values	Default
y-units	String	percent pixel	percent

- Use this parameter to specify whether you want your values for the y-coordinate to be in pixels or percent.
- If you choose percent for this parameter, and `50` for the *y-position* parameter, the click will always be in the middle of the y-axis.

2.2.1.6 Copy Text to Clipboard

`(Application)`

This action allows you to copy one string to the clipboard on your operating system. It can be used in combination with the key combination »CTRL+V« to paste text into your AUT, or into native dialogs (e.g. file choosers) opened by your AUT.

Parameters

Name	Data Type	Values	Default
Text	String	–	none

- Enter the text you wish you copy to the clipboard.
- If the text contains any special characters, use the symbol for verbatim text to mask any special functions of the characters (→ page **??**) .

2.2.1.7 Execute External Command

`(Application)`

- Use this action to run an external script or command during your test.
- You enter the command and the exit code to wait for, as well as a timeout to specify how long to wait for the code.

Parameters

Name	Data Type	Values	Default
Command	String	–	none

- Use this parameter to specify which command to execute.
- Give the path to the command, either on the local machine or on the remote machine, as defined in the *local* parameter.
- Relative paths can be written simply with the white spaces included. You will need to use quotes (") around the path or command if you are using path fragments (e.g. `./` or `../`) for the relative path.
- When using absolute paths, use quotes (") around the command or parameter containing the whitespaces.
- You need to use quotes for path fragments as
  For example, instead of:
  ```
  C:\Program Files\MyProgram\example.exe
  -data C:\Program Files\MyProgram\ws
  ```
  enter:
  ```
  "C:\Program Files\MyProgram\example.exe"
   -data "C:\Program Files\MyProgram\ws"
  ```
- In Linux, quotes may be placed around the command and the parameter. Windows cmd.exe can only accept quotes in either the parameter or the command.
- Please bear in mind that strings within the quotes are not checked for validity.

Use .cmd commands instead of .bat commands.

Name	Data Type	Values	Default
Expected Exit Code	Integer	–	0

- Scripts generally have an exit code (for example, 0 usually means that the script was successfull).
- Use this parameter to define the expected exit code.

Name	Data Type	Values	Default
Local	Boolean	true false	false

- Use this parameter to define whether the command should be run locally (on the same machine as the ITE) or remotely (on the machine where the AUT Agent is installed, and the AUT is running.).
- Set this parameter to true to run the command locally.
- Set this parameter to false to run the command remotely.

If you are running your AUT on the same machine as the ITE, th is no difference between running the command locally or remot

- For information on using relative paths to the location of the commands on local and remote machines, see the section in this document (→ page **??**) .

Name	Data Type	Values	Default
Timeout in ms	Integer	–	5000

- Enter the amount of time (in milliseconds) that should be waited for the exit code for the command.
- If the exit code is not delivered in this timeframe, the Test Step is unsuccessful.

2.2.1.8 External Input Text

`(Application)`

- Use this action to input text into dialogs that are not supported (e.g. file choosers).
- Keystrokes can be sent to these dialogs, but no check is possible to see if they have arrived.
- For this action to work, the dialog must be in focus, and the cursor must be in the text field to be filled in.

- Combine this action with the "*External Key Combination*" action to send key combinations to native dialogs to be able to work with file choosers during your test.

Native dialogs in SWT and RCP AUT's

In SWT and RCP AUT's, this action can only enter ASCII characters (large and small letters, and numbers).

To enter characters such as spaces, periods and slashes, use the action *External Key Combination*.

We recommend using the action *Copy to Clipboard* to help fill out native dialogs in SWT and RCP AUT's. The string you copy to the clipboard can then simply be entered into the textfield in the dialog using »CTRL+V« as an *External Key Combination* action.

Parameters

Name	Data Type	Values	Default
Text	String	–	none

- Use this parameter to specify the text you want to enter into the component.

2.2.1.9 External Key Combination

(Application)

- Use this action to send a key combination to a dialog that is not supported (e.g. file choosers).
- Keystrokes canbe sent to these dialogs, but it is not possible to check if it has arrived.
- For this action to work, the dialog must be in focus.
- Combine this action with the "*External Input Text*" action to be able to work with file choosers during your test.
- Some of the keystrokes you may need include *ENTER*, *SPACE*, *PERIOD* and */*.

Under Swing, this action only supports the American character set. Under SWT, you can use any character set under Windows, but there are some problems with non-American characters under Linux.

We recommend using the action *Copy to Clipboard* to help fill out native dialogs in SWT and RCP AUT's. The string you copy to the clipboard can then simply be entered into the textfield in the dialog using »Ctrl+V« as an *External Key Combination* action.

Parameters

Name	Data Type	Values	Default
Modifier Keys	String	none shift control alt meta cmd mod	none

- Use this parameter to specify a key or keys to be pressed at the same time as the action is being executed.
- You can use this parameter to specify the following keys, or combinations thereof:

  `shift`

  `control` This is the first modifier for Windows and Linux

  `cmd` This is the first modifier for Mac

  `mod` This is the platform-independent first modifier. It corresponds to »Control« for Windows and Linux, and »Cmd« for Mac.

  `meta`

  `alt`
- You can also select "*none*" if you do not need a modifier.
- You can enter multiple modifiers for a key combination by entering a space-separated list of modifiers, e.g. `alt shift`.

Do not use quotes around the modifiers, enter them in plain te

Name	Data Type	Values	Default
Base Key	String	–	none

- Use the *"base key"* parameter to specify which key to "press".
- The various keys have different codes.
- The most important keycodes are:

value(s)	description
0 ... 9	top-row keys 0 through 9
A ... Z	letters A to Z. For capital letters, use `shift` in the *"Modifier"* parameter.
ENTER	Enter or Return key
SPACE	the Spacebar
TAB	the Tab key
ESCAPE	the Escape key
BACK_SPACE	the Backspace key
F1 ... F12	the function keys
HOME, END	the home and end keys (not the number pad keys!)
INSERT, DELETE	the insertion and deletion keys (not the number pad keys!)
PAGE_UP, PAGE_DOWN	the page up and page down keys (not the number pad keys!)
DOWN, UP	the Up and Down arrow keys (not the number pad keys!)
LEFT, RIGHT	the Left and Right arrow keys (not the number pad keys!)
NUMPAD0 ... NUMPAD9	The number pad keys.

2.2.1.10 Input Text

`(Application)`

- Use this action to enter text into the component which currently has focus.
- The text you want to enter is given as a parameter.
- The difference between this action and the *"External Input Text"* is that the arrival of the keystrokes at the component are checked here.
- Use this action for supported components, and the *"External Input Text"* action for components in external dialogs, such as file choosers.

In HTML AUT's, this action can only be used with alphanumeric characters. Also, if the focus is not currently on the browser, then this action will not be successful.

Parameters

Name	Data Type	Values	Default
Text	String	–	none

- Use this parameter to specify the text you want to enter into the component.

2.2.1.11 Key Combination

`(Application)`

- This action allows you to send a key command to the application.
- This is useful for actions such as »DELETE«, »ENTER«, etc.

Under Swing, this action only supports the American character set. Under SWT, you can use any character set under Windows, but there are some problems with non-American characters under Linux.

In HTML, on Firefox, the keys »F5« and »Escape« cannot be used.

Users working on Linux systems should increase the *key repeat delay* on their test system or remove it completely. Some Linux systems can otherwise tend to repeat key presses which can lead to undesired actions being produced during test execution.

Parameters

Name	Data Type	Values	Default
Modifier Keys	String	none shift control alt meta cmd mod	none

- Use this parameter to specify a key or keys to be pressed at the same time as the action is being executed.
- You can use this parameter to specify the following keys, or combinations thereof:

  `shift`

  `control` This is the first modifier for Windows and Linux

  `cmd` This is the first modifier for Mac

  `mod` This is the platform-independent first modifier. It corresponds to »Control« for Windows and Linux, and »Cmd« for Mac.

  `meta`

  `alt`

- You can also select "*none*" if you do not need a modifier.
- You can enter multiple modifiers for a key combination by entering a space-separated list of modifiers, e.g. `alt shift`.

Do not use quotes around the modifiers, enter them in plain text

Name	Data Type	Values	Default
Base Key	String	–	none

- Use the "*base key*" parameter to specify which key to "press".
- The various keys have different codes.
- The most important keycodes are:

value(s)	**description**
0 ... 9	top-row keys 0 through 9
A ... Z	letters A to Z. For capital letters, use `shift` in the "*Modifier*" parameter.
ENTER	Enter or Return key
SPACE	the Spacebar
TAB	the Tab key
ESCAPE	the Escape key
BACK_SPACE	the Backspace key
F1 ... F12	the function keys
HOME, END	the home and end keys (not the number pad keys!)
INSERT, DELETE	the insertion and deletion keys (not the number pad keys!)
PAGE_UP, PAGE_DOWN	the page up and page down keys (not the number pad keys!)
DOWN, UP	the Up and Down arrow keys (not the number pad keys!)
LEFT, RIGHT	the Left and Right arrow keys (not the number pad keys!)
NUMPAD0 ... NUMPAD9	The number pad keys.

2.2.1.12 Manual Test Step

(Application)

- Use this action to specify a manual Test Step.
- Manual Test Steps can be specified and must be executed by a manual tester.
- There is a special test execution mode for manual testing, in which the manual Test Steps are presented to the user to be executed.

Parameters

Name	Data Type	Values	Default
Action to perform	String	–	none

- Enter a description of the action you wish to be performed.
- This description will be displayed during execution.

Name	Data Type	Values	Default
Expected Behavior	String	–	none

- Enter a description of the expected behaviour of the action.
- This description will be displayed during execution.

Name	Data Type	Values	Default
Timeout in ms	Integer	–	900000

- Enter the amount of time to wait for the execution of the manual Test Step, in milliseconds.
- If the manual Test Step is not manually passed or failed by the tester in this time, the action is marked as failed.

We recommend entering a higher timeout for manual Test St than for automated ones, as a human tester may need more ti to perform the action(s) and document any errors.

2.2.1.13 Pause Test Execution

(Application)

- This action pauses the test execution.
- To continue a paused test, press the pause button in the ITE.

This action has no parameters.

2.2.1.14 Prepare for AUT termination

`(Application)`

- Use this action together with an action to close your AUT and action to synchronize the termination and re-start of the AUT.
- Using these actions, you can close your AUT using e.g. `File` → `Exit` and have it automatically restarted.

Using the *Restart* option in Eclipse AUT's is not supported.

- Use this action shortly before an action in your test that will cause the AUT to shutdown. It must not be directly before the action to close the AUT.

The shutdown method you choose must run through the JVM shutdown hooks to properly terminate the AUT. Do not use the *restart* action to shut your AUT down.

- Using this action ensures that the step to terminate the AUT is correctly recognized and that it can be successfully re-started using the synchronize termination and re-start action. This is achieved by delaying the time between opting to close the AUT and actually closing it so that the correct communication can take place. The default delay time is set to 2000 milliseconds, and can be altered using either an environment variable (→*User Manual* p. 96) or as a JVM property in the AUT configuration (→*User Manual* p. 58) *TEST_AUT_KEEP_ALIVE_DELAY*. Any test step delay specified in your Test Suite is added to this delay.

This action is currently supported on the Swing and RCP toolkits. Neither toolkit supports the terminating of AUT's using the *press any key* actions to press e.g. »Alt+F4«. External key combinations can be used to terminate the AUT using e.g. »Alt+F4« on RCP AUT's, but not Swing AUT's.

This action has no parameters.

2.2.1.15 Read Timer

`(Application)`

- Use this action to read a timer (stopwatch).
- You enter the name of the timer you want to read and enter a variable which saves the difference (in milliseconds) between the current time and the time at which the timer was started.
- You can start timers and compare values between starting and reading timers using other actions.
- This action is useful for measuring how long your AUT takes to execute certain actions.

You must have used the action*Start timer* at least once in your test to use this action.

Parameters

Name	Data Type	Values	Default
Timer Name	String	–	none

- Enter the name of the timer you want to read.

Name	Data Type	Values	Default
Timer Value Variable Name	Variable	–	none

- Enter the variable name you will use to save the time difference between starting the timer and reading it.

2.2.1.16 Restart

`(Application)`

- Use this action to restart the AUT during the test.
- The action closes your AUT and all windows/dialogs associated with it.
- It then starts the same AUT with the same configuration as it had when you originally started it and carries on with the test.
- If your AUT takes a while to load/to be ready, it is a good idea to use the *"wait for window"* or *"wait for component"* action after the restart action.

Do not use the restart action as the first Test Step in a test.

If you want to close your AUT using e.g. the menu and have it restarted automatically, then use the actions to prepare for AUT termination and to synchronize termination and re-start of an AUT.

This action has no parameters.

2.2.1.17 Set Toggle Key

`(Application)`

- Use this action to turn the following keys on or off:
  - Caps-lock
  - Num-lock
  - Scroll-lock
- You specify which key to (de)activate and whether the function should be turned on or off.

Due to current Java limitations, this action is not supported when testing under Linux. This means that any Test Step containing this action will fail when testing under Linux.

Parameters

Name	Data Type	Values	Default
Key	Integer	1 2 3	1

- Use this parameter to specify which key you want to (de)activate.
- `1` = Num-lock
- `2` = Caps-lock
- `3` = Scroll-lock

Name	Data Type	Values	Default
Activate Key	Boolean	true false	true

- Use this parameter to specify whether you want to turn the function on or off.
- Select true to turn the function on.
- Select false to turn the function off.
- Turning the function on when it is already on will not have any effect. The function remains on.

2.2.1.18 Start Timer

`(Application)`

- Use this action to start a timer (stopwatch).
- You name the timer and enter a variable which saves the current time (in milliseconds since the January 1st 1970).
- You can read timers and compare values between starting and reading timers using other actions.
- This action is useful for measuring how long your AUT takes to execute certain actions.

Parameters

Name	Data Type	Values	Default
Timer Name	String	–	none

- Enter a name you will use to identify this timer in your tests.

Name	Data Type	Values	Default
Start Time Variable Name	Variable	–	none

- Enter the variable name you will use to save the current time.

2.2.1.19 Store Value

`(Application)`

- Use this action to store a value you specify so that you can use it as data for other actions.

Parameters

Name	Data Type	Values	Default
Variable Name	Variable	–	none

- Enter the name you want to give to this variable.
- Variable names may only contain letters, numbers and underscores.
- You can then enter this variable name as data for other Test Steps.
- When you enter the variable name as data, place a dollar sign before it.
- The data associated with this variable name remains the same until the Project is closed, or until you overwrite the name by using it for another value.

Name	Data Type	Values	Default
Value	String	–	none

- Enter the value that you want to store.

2.2.1.20 Synchronize termination and re-start of AUT

`(Application)`

- Use this action together with the action to prepare for AUT termination and an action to close your AUT.
- Using these actions, you can close your AUT using e.g. `File` → `Exit` and have it automatically restarted.

Using the *Restart* option in Eclipse AUT's is not supported.

- Use this action sometime after using the prepare for termination action, and directly after the action to close the AUT.

The shutdown method you choose must run through the JVM shutdown hooks to properly terminate the AUT. Do not use the *restart* action to shut your AUT down.

- The action works by waiting for the AUT to terminate (configurable via a parameter), then re-starting the AUT using the same configuration / autrun parameters as it was originally started with.
- You may want to follow this action with a module that waits for your AUT to be started completely.

This action is currently supported on the Swing and RCP toolkits. Neither toolkit supports the terminating of AUT's using the *press any key* actions to press e.g. »Alt+F4«. External key combinations can be used to terminate the AUT using e.g. »Alt+F4« on RCP AUT's, but not Swing AUT's.

Parameters

Name	Data Type	Values	Default
Timeout in ms	Integer	–	10000

- Specify the timeout in milliseconds that represents the maximum amount of time that should be waited for the AUT to terminate (to de-register from the AUT Agent). You should specify a time that represents how long your AUT takes to close.
- If, after closing, your AUT requires time to free up resources, save settings etc., then you can use the environment variable *TEST_AUT_POST_DEREGISTRATION_DELAY* for the environment the AUT Agent is running in (→*User Manual* p. 96) or as a JVM property in the AUT configuration (→*User Manual* p. 58) to add a delay to ensure that these steps are completed successfully.
- If the termination timeout is exceeded, and the AUT is still registered, then a normal *restart* AUT will be executed and this Test Step will be marked as failed.

The timestamp for this Test Step will usually be longer than the timeout specified, due to internal actions that are necessary to synchronize the termination and re-start.

This action is currently supported on the Swing and RCP toolkits.

2.2.1.21 Take Screenshot

`(Application)`

- This action takes a screen capture and saves the resulting image to disk.
- The action takes a screenshot of the whole screen on the primary monitor. To take a screenshot of only the active window, use the action *Take Screenshot of Active Window*.
- If you do not enter a file extension, the screenshot will be saved as a *.png* file. You can also save the file as an *.jpeg* or *.bmp* file.

Parameters

Name	Data Type	Values	Default
Destination	String	–	none

- Use this parameter to define the path and filename of the saved image.
- Make sure that you have write access to the destination.

- If the destination file does not have an extension, the *.png*extension will be automatically appended.
- For information on using relative paths to the location where screenshots should be saved, see the section in this document (→ page **??**) .

Name	Data Type	Values	Default
Delay	Integer	–	0

- Use this parameter to insert a delay before taking the screenshot.

Name	Data Type	Values	Default
File Access	String	rename overwrite	rename

- Use this parameter to control how this image is saved if the destination file already exists.
- Select ”overwrite” to simply overwrite the Destination file.
- Select ”rename” to append a sequential integer to the file name. This automates the creation of sequentially labeled screenshots.

Name	Data Type	Values	Default
Scaling Factor	Integer	–	100

- Use this parameter to scale the image created by the screenshot.
- The value for this parameter must lie between 1 and 200 (inclusive).
- The units for this parameter are ”percent”. This means that a value of 50 scales the image to half its original size, and a value of 200 scales the image to double its original size.

Name	Data Type	Values	Default
Create Directories	Boolean	true false	true

- Use this parameter to determine how the image is saved if the given path does not exist.
- Select true to create the necessary directories automatically.
- Selecting false will cause the test step to fail if the path does not already exist.

2.2.1.22 Take Screenshot of Active Window

`(Application)`

- This action takes a screen capture of the active window on the primary monitor and saves the resulting image to disk.
- To take a screenshot of thw whole screen, use the action *Take Screenshot*.
- If you do not enter a file extension, the screenshot will be saved as a *.png* file. You can also save the file as an *.jpeg* or *.bmp* file.
- You can specify a margin around the active window to extend the range of the screenshot. This may be necessary on some systems where the title bar of some dialogs is not included in the screenshot.

Parameters

Name	Data Type	Values	Default
Destination	String	–	none

- Use this parameter to define the path and filename of the saved image.
- Make sure that you have write access to the destination.
- If the destination file does not have an extension, the *.png*extension will be automatically appended.
- For information on using relative paths to the location where screenshots should be saved, see the section in this document (→ page **??**) .

Name	Data Type	Values	Default
Delay	Integer	–	0

- Use this parameter to insert a delay before taking the screenshot.

Name	Data Type	Values	Default
File Access	String	rename overwrite	rename

- Use this parameter to control how this image is saved if the destination file already exists.
- Select ”overwrite” to simply overwrite the Destination file.
- Select ”rename” to append a sequential integer to the file name. This automates the creation of sequentially labeled screenshots.

Name	Data Type	Values	Default
Scaling Factor	Integer	–	100

- Use this parameter to scale the image created by the screenshot.
- The value for this parameter must lie between 1 and 200 (inclusive).
- The units for this parameter are "percent". This means that a value of 50 scales the image to half its original size, and a value of 200 scales the image to double its original size.

Name	Data Type	Values	Default
Create Directories	Boolean	true false	true

- Use this parameter to determine how the image is saved if the given path does not exist.
- Select true to create the necessary directories automatically.
- Selecting false will cause the test step to fail if the path does not already exist.

Name	Data Type	Values	Default
Margin Top	Integer	–	0

- Use this parameter to enter the amount of pixels you wish you include in the screenshot above the top of the dialog.
- You can enter positive and negative values.
- Enter 0 to neither extend nor reduce the margin at the top of the dialog.

Name	Data Type	Values	Default
Margin Right	Integer	–	0

- Use this parameter to enter the amount of pixels you wish you include in the screenshot to the right of the dialog.
- You can enter positive and negative values.
- Enter 0 to neither extend nor reduce the margin.

Name	Data Type	Values	Default
Margin Bottom	Integer	–	0

- Use this parameter to enter the amount of pixels you wish you include in the screenshot at the bottom of the dialog.
- You can enter positive and negative values.
- Enter 0 to neither extend nor reduce the margin.

Name	Data Type	Values	Default
Margin Left	Integer	–	0

- Use this parameter to enter the amount of pixels you wish you include in the screenshot at left of the dialog.
- You can enter positive and negative values.
- Enter 0 to neither extend nor reduce the margin.

2.2.1.23 Wait

`(Application)`

- Sometimes it is necessary to wait for a period of time before continuing.
- Use this action to tell the AUT Agent to wait for a given amount of time before continuing with the next Test Step.
- There are also actions to wait for a window and a component. If you are waiting for a component, or a window, use the actions specific to these events.

Parameters

Name	Data Type	Values	Default
Millisecs	Integer	–	none

- Use this parameter to enter the amount of time to wait before carrying on.
- Enter the amount of time in milliseconds.

2.2.1.24 Wait for Window

`(Application)`

- This action waits for the window you name to open.
- This action does *not* check whether the window is active
- Once the window is open (or if the window is already open), the test continues.
- If the window does not open, the action fails.

Parameters

Name	Data Type	Values	Default
Title	String	–	none

- Enter the title of the window you are waiting for.
- The title you provide will be compared to the title bar of each window as it opens, using the Operator provided.

Name	Data Type	Values	Default
Operator	String	equals not equals matches simple match	equals

- The *operator* parameter has four possible values.
- "*not equals*" looks for something that does *not exactly* match.
- "*equals*" looks for an *exact* match.
- Select "*simple match*" to use a simple match expression (→ page **??**) .
- Select "*matches*" to use a regular expression (→ page **??**) .
- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

Name	Data Type	Values	Default
Timeout in ms	Integer	–	1000

- Enter the amount of time to wait for the window, in milliseconds.
- If the window does not open in this time, the action fails.

Name	Data Type	Values	Default
Delay after Visibility	Integer	–	200

- Sometimes a window is not ready to receive events (mouse clicks, key presses) until a few moments after becoming visible.
- Use this parameter to set the amount of time (in milliseconds) to wait once the window has appeared before continuing.

2.2.1.25 Wait for Window Activation

`(Application)`

- This action waits for the window you name to become active. A window is active only when it is the currently selected window.
- Once the window is active (or if the window is already active when this test step begins), the test continues.
- If the window does not become active, the action fails.

Parameters

Name	Data Type	Values	Default
Title	String	–	none

- Enter the title of the window you are waiting for.
- Enter the title exactly as it appears in the title bar of the window.
- The title you provide will be compared to the title bar of each window as it becomes active, using the Operator provided.

Name	Data Type	Values	Default
Operator	String	equals not equals matches simple match	equals

- The *operator* parameter has four possible values.
- "*not equals*" looks for something that does *not exactly* match.
- "*equals*" looks for an *exact* match.
- Select "*simple match*" to use a simple match expression (→ page **??**) .
- Select "*matches*" to use a regular expression (→ page **??**) .
- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

Name	Data Type	Values	Default
Timeout in ms	Integer	–	1000

- Enter the amount of time to wait for the window, in milliseconds.
- If the window does not appear in this time, the action fails.

Name	Data Type	Values	Default
Delay after Visibility	Integer	–	200

- Sometimes a window is not ready to receive events (mouse clicks, key presses) until a few moments after activation.
- Use this parameter to set the amount of time (in milliseconds) to wait once the window has appeared before continuing.

2.2.1.26 Wait for Window to Close

`(Application)`

- This action waits for the window you name to close.
- Once the window is closed (or if no open window with a corresponding title can be found), the test continues.
- If a window is found and does not close within the timeout period, the action fails.

Parameters

Name	Data Type	Values	Default
Title	String	–	none

- Enter the title of the window you are waiting for.
- Enter the title exactly as it appears in the title bar of the window.
- The title you provide will be compared to the title bar of each window as it closes, using the Operator provided.

Name	Data Type	Values	Default
Operator	String	equals not equals matches simple match	equals

- The *operator* parameter has four possible values.
- "*not equals*" looks for something that does *not exactly* match.
- "*equals*" looks for an *exact* match.
- Select "*simple match*" to use a simple match expression (→ page **??**) .
- Select "*matches*" to use a regular expression (→ page **??**) .
- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

Name	Data Type	Values	Default
Timeout in ms	Integer	–	1000

- Enter the amount of time to wait for the window, in milliseconds.
- If the window does not close in this time, the action fails.

Name	Data Type	Values	Default
Delay after Closure	Integer	–	200

- Sometimes other windows are not ready to receive events (mouse clicks, key presses) until a few moments after becoming the overlapping window is closed.
- Use this parameter to set the amount of time (in milliseconds) to wait once the window has closed before continuing.

2.2.2 Button Component

Description:

- This component includes standard buttons, check boxes and radio buttons.
- A *Button* is a "push" button, usually manipulated with a click of the mouse over the button area:

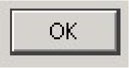


Figure 2.1: *Button*

- A *Check Box* is commonly used to select or unselect an option (toggle):

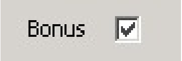


Figure 2.2: *Checkbox*

- A *Radio Button* is used to choose one of a list of options:

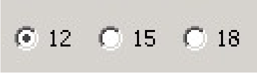


Figure 2.3: *Radio Button*

Synopsis:

- Button Component (concrete)
  - Button Component (abstract)
    - Text Component (abstract)
      - Graphics Component (abstract)

New Actions

none

Inherited Actions

Name	Type	Parameters	Inherited from
Check Enablement (→ page 23)	c	Boolean:Enabled	Graphics Component (abstract)

Check Enablement of Context Menu Entry by Indexpath (→ page 24)	c	String:Indexpath Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)
Check Enablement of Context Menu Entry by Indexpath (Specify Position) (→ page 25)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Indexpath Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)
Check Enablement of Context Menu Entry by Textpath (→ page 28)	c	String:Textpath String:Operator Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)
Check Enablement of Context Menu Entry by Textpath (Specify Position) (→ page 29)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Textpath String:Operator Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)
Check Existence (→ page 32)	c	Boolean:Exists	Graphics Component (abstract)
Check Existence of Context Menu Entry by Indexpath (→ page 33)	c	String:Indexpath Boolean:Exists Integer:Mouse Button	Graphics Component (abstract)
Check Existence of Context Menu Entry by Indexpath (Specify Position) (→ page 34)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Indexpath Boolean:Exists Integer:Mouse Button	Graphics Component (abstract)

Check Existence of Context Menu Entry by Textpath (→ page 37)	c	String:Textpath String:Operator Boolean:Exists Integer:Mouse Button	Graphics Component (abstract)
Check Existence of Context Menu Entry by Textpath (Specify Position) (→ page 38)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Textpath String:Operator Boolean:Exists Integer:Mouse Button	Graphics Component (abstract)
Check Focus (→ page 41)	c	Boolean:Has Focus	Graphics Component (abstract)
Check Property (→ page 42)	c	String:Property Name String:Property Value String:Operator	Graphics Component (abstract)
Check Selection (→ page 19)	c	Boolean:Selected	Button Component (abstract)
Check Selection of Context Menu Entry by Indexpath (→ page 43)	c	String:Indexpath Boolean:Selected Integer:Mouse Button	Graphics Component (abstract)
Check Selection of Context Menu Entry by Indexpath (Specify Position) (→ page 45)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Indexpath Boolean:Selected Integer:Mouse Button	Graphics Component (abstract)

Check Selection of Context Menu Entry by Textpath (→ page 47)	c	String:Textpath String:Operator Boolean:Selected Integer:Mouse Button	Graphics Component (abstract)
Check Selection of Context Menu Entry by Textpath (Specify Position) (→ page 49)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Textpath String:Operator Boolean:Selected Integer:Mouse Button	Graphics Component (abstract)
Check Text (→ page 76)	c	String:Text String:Operator	Text Component (abstract)
Click (→ page 52)	e	Integer:Number of Clicks Integer:Mouse Button	Graphics Component (abstract)
Click in Component (→ page 53)	e	Integer:Number of Clicks Integer:Mouse Button Integer:x-position String:x-units Integer:y-position String:y-units	Graphics Component (abstract)
Drag (→ page 55)	e	Integer:Mouse Button String:Modifier Keys Integer:x-position String:x-units Integer:y-position String:y-units	Graphics Component (abstract)

Drop (→ page 58)	e	Integer:x-position String:x-units Integer:y-position String:y-units Integer:Delay before drop (milliseconds)	Graphics Component (abstract)
Select Context Menu Entry by Indexpath (→ page 60)	e	String:Indexpath Integer:Mouse Button	Graphics Component (abstract)
Select Context Menu Entry by Indexpath (Specify Position) (→ page 61)	e	Integer:x-position String:x-units Integer:y-position String:y-units String:Indexpath Integer:Mouse Button	Graphics Component (abstract)
Select Context Menu Entry by Textpath (→ page 64)	e	String:Textpath String:Operator Integer:Mouse Button	Graphics Component (abstract)
Select Context Menu Entry by Textpath (Specify Position) (→ page 66)	e	Integer:x-position String:x-units Integer:y-position String:y-units String:Menupath String:Operator Integer:Mouse Button	Graphics Component (abstract)

Show Text (→ page 69)	e	String:Text Integer:Text Size (in points) Integer:Time per Word (in milliseconds) Integer:Window Width (in pixels)	Graphics Component (abstract)
Store Property (→ page 70)	e	Variable:Variable Name String:Property Name	Graphics Component (abstract)
Store Value (→ page 77)	e	Variable:Variable Name	Text Component (abstract)
Wait for Component (→ page 71)	e	Integer:Timeout in ms Integer:Delay after Visibility	Graphics Component (abstract)

Used By

(SWT) Tool Item (swt)

2.2.3 Combo Component

Description:

- A *combo box* is a component which consists of a field or button and a drop down list:

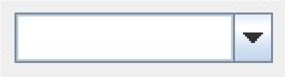


Figure 2.4: *Combo Box*

- When the field or button is clicked, the list opens, allowing you to select one of a number of values.

Because the comma (,) is a special symbol for combo boxes, if you want to use a comma as part of your parameter value, you have to mask it. See the section later in this document (→ page **??**) for more details.

Mapping combo boxes

In the Object Mapping Mode, a combo box to be mapped looks like this:

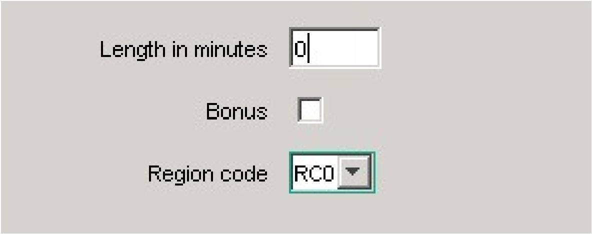


Figure 2.5: *Combo Box*

If you execute a *click in component* or *click* action on a combo box, please be aware that the combo box will then be active, and must be deactivated (e.g. by pressing escape) before continuing with other steps (e.g. clicking buttons) in the test.

Synopsis:

- Combo Component (concrete)
  - Text Input Component (abstract)
    - Text Component (abstract)
      - Graphics Component (abstract)

New Actions

Name	Type	Parameters
Check Existence of Entry by Value (→ page 131)	c	String:Text String:Operator Boolean:Exists
Check Selection of Entry by Index (→ page 132)	c	String:Index Boolean:Selected
Select Entry by Index (→ page 133)	e	String:Index
Select Entry by Value (→ page 133)	e	String:Text String:Operator String:Search Type

Inherited Actions

Name	Type	Parameters	Inherited from
Check Editability (→ page 84)	c	Boolean:Editable	Text Input Component (abstract)
Check Enablement (→ page 23)	c	Boolean:Enabled	Graphics Component (abstract)
Check Enablement of Context Menu Entry by Indexpath (→ page 24)	c	String:Indexpath Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)
Check Enablement of Context Menu Entry by Indexpath (Specify Position) (→ page 25)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Indexpath Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)
Check Enablement of Context Menu Entry by Textpath (→ page 28)	c	String:Textpath String:Operator Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)
Check Enablement of Context Menu Entry by Textpath (Specify Position) (→ page 29)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Textpath String:Operator Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)
Check Existence (→ page 32)	c	Boolean:Exists	Graphics Component (abstract)
Check Existence of Context Menu Entry by Indexpath (→ page 33)	c	String:Indexpath Boolean:Exists Integer:Mouse Button	Graphics Component (abstract)

Check Existence of Context Menu Entry by Indexpath (Specify Position) (→ page 34)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Indexpath Boolean:Exists Integer:Mouse Button	Graphics Component (abstract)
Check Existence of Context Menu Entry by Textpath (→ page 37)	c	String:Textpath String:Operator Boolean:Exists Integer:Mouse Button	Graphics Component (abstract)
Check Existence of Context Menu Entry by Textpath (Specify Position) (→ page 38)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Textpath String:Operator Boolean:Exists Integer:Mouse Button	Graphics Component (abstract)
Check Focus (→ page 41)	c	Boolean:Has Focus	Graphics Component (abstract)
Check Property (→ page 42)	c	String:Property Name String:Property Value String:Operator	Graphics Component (abstract)
Check Selection of Context Menu Entry by Indexpath (→ page 43)	c	String:Indexpath Boolean:Selected Integer:Mouse Button	Graphics Component (abstract)
Check Selection of Context Menu Entry by Indexpath (Specify Position) (→ page 45)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Indexpath Boolean:Selected Integer:Mouse Button	Graphics Component (abstract)

Check Selection of Context Menu Entry by Textpath (→ page 47)	c	String:Textpath String:Operator Boolean:Selected Integer:Mouse Button	Graphics Component (abstract)
Check Selection of Context Menu Entry by Textpath (Specify Position) (→ page 49)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Textpath String:Operator Boolean:Selected Integer:Mouse Button	Graphics Component (abstract)
Check Text (→ page 76)	c	String:Text String:Operator	Text Component (abstract)
Click (→ page 52)	e	Integer:Number of Clicks Integer:Mouse Button	Graphics Component (abstract)
Click in Component (→ page 53)	e	Integer:Number of Clicks Integer:Mouse Button Integer:x-position String:x-units Integer:y-position String:y-units	Graphics Component (abstract)
Drag (→ page 55)	e	Integer:Mouse Button String:Modifier Keys Integer:x-position String:x-units Integer:y-position String:y-units	Graphics Component (abstract)

Drop (→ page 58)	e	Integer:x-position String:x-units Integer:y-position String:y-units Integer:Delay before drop (milliseconds)	Graphics Component (abstract)
Input Text (→ page 84)	e	String:Text	Text Input Component (abstract)
Replace Text (→ page 85)	e	String:Text	Text Input Component (abstract)
Select Context Menu Entry by Indexpath (→ page 60)	e	String:Indexpath Integer:Mouse Button	Graphics Component (abstract)
Select Context Menu Entry by Indexpath (Specify Position) (→ page 61)	e	Integer:x-position String:x-units Integer:y-position String:y-units String:Indexpath Integer:Mouse Button	Graphics Component (abstract)
Select Context Menu Entry by Textpath (→ page 64)	e	String:Textpath String:Operator Integer:Mouse Button	Graphics Component (abstract)
Select Context Menu Entry by Textpath (Specify Position) (→ page 66)	e	Integer:x-position String:x-units Integer:y-position String:y-units String:Menupath String:Operator Integer:Mouse Button	Graphics Component (abstract)

Show Text (→ page 69)	e	String:Text Integer:Text Size (in points) Integer:Time per Word (in milliseconds) Integer:Window Width (in pixels)	Graphics Component (abstract)
Store Property (→ page 70)	e	Variable:Variable Name String:Property Name	Graphics Component (abstract)
Store Value (→ page 77)	e	Variable:Variable Name	Text Component (abstract)
Wait for Component (→ page 71)	e	Integer:Timeout in ms Integer:Delay after Visibility	Graphics Component (abstract)

2.2.3.1 Check Existence of Entry by Value

```
(Combo Component)
```

- Use this action to check whether a value you enter is contained in the combo box.

Parameters

Name	Data Type	Values	Default
Text	String	–	none

- Enter the value you want to check for.
- This parameter will be compared to entries in the Combo Box, using the Operator provided.

Name	Data Type	Values	Default
Operator	String	equals not equals matches simple match	equals

- The *operator* parameter has four possible values.
- "*not equals*" looks for something that does *not exactly* match.
- "*equals*" looks for an *exact* match.

- Select *"simple match"* to use a simple match expression (→ page **??**) .
- Select *"matches"* to use a regular expression (→ page **??**) .
- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

Name	Data Type	Values	Default
Exists	Boolean	true false	true

- Set this parameter to *true* if you expect the entry to exist.
- Set this parameter to *false* if you expect the entry not to exist.

2.2.3.2 Check Selection of Entry by Index

`(Combo Component)`

- Use this action to test whether the item at the given index is currently selected in the combo box.

Parameters

Name	Data Type	Values	Default
Index	String	–	none

- Enter the index of the item you want to check.
- If you want to check that the first item is selected, enter `1`.
- If you want to check that the fifth item is selected, enter `5`.

Name	Data Type	Values	Default
Selected	Boolean	true false	true

- Set this parameter to *true* if you expect the entry to be selected.
- Set this parameter to *false* if you expect the entry not to be selected.

2.2.3.3 Select Entry by Index

`(Combo Component)`

- Use this action to select an item from the combo box using its index as a parameter.
- This is achieved by:
  - Clicking once on the combo box to open the list.
  - Selecting the item specified by clicking it once.

Parameters

Name	Data Type	Values	Default
Index	String	–	none

- Use this parameter to specify which item you want to select from the combo box.
- Refer to the item using its index.
- The first item has an index of `1`, the second `2` and so on.

2.2.3.4 Select Entry by Value

`(Combo Component)`

- Use this action to select an item from the combo box using the name of the item as a parameter.
- This is achieved by:
  - Clicking once on the combo box to open the list.
  - Selecting the item specified by clicking it once.

Parameters

Name	Data Type	Values	Default
Text	String	–	none

- Use this parameter to enter the value you want to select.
- This parameter will be compared to entries in the Combo Box, using the Operator provided.

Name	Data Type	Values	Default
Operator	String	equals not equals matches simple match	equals

- The *operator* parameter has four possible values.
- "*not equals*" looks for something that does *not exactly* match.
- "*equals*" looks for an *exact* match.
- Select "*simple match*" to use a simple match expression (→ page **??**) .
- Select "*matches*" to use a regular expression (→ page **??**) .
- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

Name	Data Type	Values	Default
Search Type	String	absolute relative	absolute

- The *search type* parameter has two options.
- "*absolute*" begins searching for the value at the first entry.
- Select "*relative*" to begin the search for the value at the currently selected entry. This allows you to search for the next occurence of a value, ignoring all elements before the selected entry.

2.2.4 List Component

Description:

A *List* is a component which displays a number of values. Lists allow single or multiple values to be selected.

Because the comma (,) is a special symbol for lists, if you want to use a comma as part of your parameter value, you have to mask it. See the section later in this document (→ page **??**) for more details.

Figure 2.6: *List*

Mapping lists

In the Object Mapping Mode, a list to be mapped looks like this:

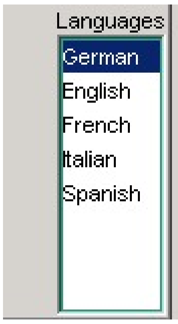


Figure 2.7: *List*

Synopsis:

- List Component (concrete)
  - Text Component (abstract)
    - Graphics Component (abstract)

New Actions

Name	Type	Parameters
Check Existence of Entry by Value (→ page 141)	c	String:Text String:Operator Boolean:Exists
Check Selection of Entry by Index (→ page 142)	c	String:Index Boolean:Selected
Check Selection of Entry by Value (→ page 143)	c	String:Text String:Operator Boolean:Selected
Drag Entry by Index (→ page 144)	e	Integer:Mouse Button String:Modifier Keys Integer:Index
Drag Entry by Value (→ page 145)	e	Integer:Mouse Button String:Modifier Keys String:Text String:Operator String:Search Type
Drop on Entry by Index (→ page 147)	e	Integer:Index Integer:Delay before drop (milliseconds)
Drop on Entry by Value (→ page 148)	e	String:Text String:Operator String:Search Type Integer:Delay before drop (milliseconds)
Select Entry by Index/Indices (→ page 150)	e	String:Index String:Extend Selection Integer:Mouse Button Integer:Number of Clicks
Select Entry by Value(s) (→ page 151)	e	String:Text String:Operator String:Search Type String:Extend Selection Integer:Mouse Button Integer:Number of Clicks

Inherited Actions

Name	Type	Parameters	Inherited from
Check Enablement (→ page 23)	c	Boolean:Enabled	Graphics Component (abstract)
Check Enablement of Context Menu Entry by Indexpath (→ page 24)	c	String:Indexpath Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)
Check Enablement of Context Menu Entry by Indexpath (Specify Position) (→ page 25)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Indexpath Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)
Check Enablement of Context Menu Entry by Textpath (→ page 28)	c	String:Textpath String:Operator Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)
Check Enablement of Context Menu Entry by Textpath (Specify Position) (→ page 29)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Textpath String:Operator Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)

Check Existence (→ page 32)	c	Boolean:Exists	Graphics Component (abstract)
Check Existence of Context Menu Entry by Indexpath (→ page 33)	c	String:Indexpath Boolean:Exists Integer:Mouse Button	Graphics Component (abstract)
Check Existence of Context Menu Entry by Indexpath (Specify Position) (→ page 34)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Indexpath Boolean:Exists Integer:Mouse Button	Graphics Component (abstract)
Check Existence of Context Menu Entry by Textpath (→ page 37)	c	String:Textpath String:Operator Boolean:Exists Integer:Mouse Button	Graphics Component (abstract)
Check Existence of Context Menu Entry by Textpath (Specify Position) (→ page 38)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Textpath String:Operator Boolean:Exists Integer:Mouse Button	Graphics Component (abstract)
Check Focus (→ page 41)	c	Boolean:Has Focus	Graphics Component (abstract)
Check Property (→ page 42)	c	String:Property Name String:Property Value String:Operator	Graphics Component (abstract)

Check Selection of Context Menu Entry by Indexpath (→ page 43)	c	String:Indexpath Boolean:Selected Integer:Mouse Button	Graphics Component (abstract)
Check Selection of Context Menu Entry by Indexpath (Specify Position) (→ page 45)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Indexpath Boolean:Selected Integer:Mouse Button	Graphics Component (abstract)
Check Selection of Context Menu Entry by Textpath (→ page 47)	c	String:Textpath String:Operator Boolean:Selected Integer:Mouse Button	Graphics Component (abstract)
Check Selection of Context Menu Entry by Textpath (Specify Position) (→ page 49)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Textpath String:Operator Boolean:Selected Integer:Mouse Button	Graphics Component (abstract)
Check Text (→ page 76)	c	String:Text String:Operator	Text Component (abstract)
Click (→ page 52)	e	Integer:Number of Clicks Integer:Mouse Button	Graphics Component (abstract)
Click in Component (→ page 53)	e	Integer:Number of Clicks Integer:Mouse Button Integer:x-position String:x-units Integer:y-position String:y-units	Graphics Component (abstract)

Drag (→ page 55)	e	Integer:Mouse Button String:Modifier Keys Integer:x-position String:x-units Integer:y-position String:y-units	Graphics Component (abstract)
Drop (→ page 58)	e	Integer:x-position String:x-units Integer:y-position String:y-units Integer:Delay before drop (milliseconds)	Graphics Component (abstract)
Select Context Menu Entry by Indexpath (→ page 60)	e	String:Indexpath Integer:Mouse Button	Graphics Component (abstract)
Select Context Menu Entry by Indexpath (Specify Position) (→ page 61)	e	Integer:x-position String:x-units Integer:y-position String:y-units String:Indexpath Integer:Mouse Button	Graphics Component (abstract)
Select Context Menu Entry by Textpath (→ page 64)	e	String:Textpath String:Operator Integer:Mouse Button	Graphics Component (abstract)
Select Context Menu Entry by Textpath (Specify Position) (→ page 66)	e	Integer:x-position String:x-units Integer:y-position String:y-units String:Menupath String:Operator Integer:Mouse Button	Graphics Component (abstract)

Show Text (→ page 69)	e	String:Text Integer:Text Size (in points) Integer:Time per Word (in milliseconds) Integer:Window Width (in pixels)	Graphics Component (abstract)
Store Property (→ page 70)	e	Variable:Variable Name String:Property Name	Graphics Component (abstract)
Store Value (→ page 77)	e	Variable:Variable Name	Text Component (abstract)
Wait for Component (→ page 71)	e	Integer:Timeout in ms Integer:Delay after Visibility	Graphics Component (abstract)

2.2.4.1 Check Existence of Entry by Value

`(List Component)`

- This action tests whether a value you specify is contained in the list.

Parameters

Name	Data Type	Values	Default
Text	String	–	none

- Use this parameter to enter the value you want to check for.
- This parameter will be compared to entries in the List, using the Operator provided.

Name	Data Type	Values	Default
Operator	String	equals not equals matches simple match	equals

- The *operator* parameter has four possible values.
- "*not equals*" looks for something that does *not exactly* match.
- "*equals*" looks for an *exact* match.

- Select "*simple match*" to use a simple match expression (→ page **??**) .
- Select "*matches*" to use a regular expression (→ page **??**) .
- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

Name	Data Type	Values	Default
Exists	Boolean	true false	true

- Set this parameter to *true* if you expect this entry to exist.
- Set this parameter to *false* if you expect this entry not to exist.

2.2.4.2 Check Selection of Entry by Index

`(List Component)`

- Use this action to test whether an item in a list is currently selected.
- The item you want to test is given using its index.

Parameters

Name	Data Type	Values	Default
Index	String	–	none

- Enter the index for the item you want to check.
- For example, if you want to check that the first value is selected, enter `1`.
- The first item is always `1`.

Name	Data Type	Values	Default
Selected	Boolean	true false	true

- Set this parameter to *true* if you expect this entry to be selected.
- Set this parameter to *false* if you expect this entry not to be selected.

2.2.4.3 Check Selection of Entry by Value

`(List Component)`

- Use this action to test whether an item in a list is currently selected.
- The item you want to test is given using its value (name).

Parameters

Name	Data Type	Values	Default
Text	String	–	none

- Enter the name of the item you want to check.
- For example, if you want to check that "*England*" and "*Germany*" are selected, enter `England` in one Data Set and `Germany` in the next.

Name	Data Type	Values	Default
Operator	String	equals not equals matches simple match	equals

- The *operator* parameter has four possible values.
- "*not equals*" looks for something that does *not exactly* match.
- "*equals*" looks for an *exact* match.
- Select "*simple match*" to use a simple match expression (→ page **??**) .
- Select "*matches*" to use a regular expression (→ page **??**) .
- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

Name	Data Type	Values	Default
Selected	Boolean	true false	true

- Set this parameter to *true* if you expect this entry to be selected.
- Set this parameter to *false* if you expect this entry not to be selected.

2.2.4.4 Drag Entry by Index

`(List Component)`

- This action drags the list item you specify.
- The item is specified by its index in the list.
- The cursor is moved over the component.
- The mouse button you specify is held, and the item is dragged.
- Follow this action with a *drop* action

If you want to drag multiple items, use an action to select the items you want to drag before using the drag action.

Parameters

Name	Data Type	Values	Default
Mouse Button	Integer	1 2 3	1

- Use this parameter to select which mouse button you want to click with.
  - `1` = left mouse button
  - `2` = middle mouse button
  - `3` = right mouse button
- The mouse button you use to click with can also have an effect on what happens.
- The button to click will depend on your AUT and what the click should achieve.

Do not enter quotes around the mouse button numbers, enter them in plain text.

Name	Data Type	Values	Default
Modifier Keys	String	none shift control alt meta cmd mod	none

- Use this parameter to specify a key or keys to be pressed at the same time as the action is being executed.
- You can use this parameter to specify the following keys, or combinations thereof:

  `shift`

  `control` This is the first modifier for Windows and Linux

  `cmd` This is the first modifier for Mac

  `mod` This is the platform-independent first modifier. It corresponds to »CONTROL« for Windows and Linux, and »CMD« for Mac.

  `meta`

  `alt`
- You can also select "*none*" if you do not need a modifier.
- You can enter multiple modifiers for a key combination by entering a space-separated list of modifiers, e.g. `alt shift`.

Do not use quotes around the modifiers, enter them in plain te

Name	Data Type	Values	Default
Index	Integer	–	none

- Use this parameter to enter the item you want to drag.
- Use the index of the item to indicate which item to drag.
- For example, if you want to drag the first item from the list, enter: `1`.

2.2.4.5 Drag Entry by Value

`(List Component)`

- This action drags the list item you specify.
- The item is specified by its value in the list.
- The cursor is moved over the component.
- The mouse button you specify is held, and the item is dragged.
- Follow this action with a *drop* action.

If you want to drag multiple items, use an action to select the items you want to drag before using the drag action.

Parameters

Name	Data Type	Values	Default
Mouse Button	Integer	1 2 3	1

- Use this parameter to select which mouse button you want to click with.
  - `1` = left mouse button
  - `2` = middle mouse button
  - `3` = right mouse button
- The mouse button you use to click with can also have an effect on what happens.
- The button to click will depend on your AUT and what the click should achieve.

Do not enter quotes around the mouse button numbers, enter them in plain text.

Name	Data Type	Values	Default
Modifier Keys	String	none shift control alt meta cmd mod	none

- Use this parameter to specify a key or keys to be pressed at the same time as the action is being executed.
- You can use this parameter to specify the following keys, or combinations thereof:

  `shift`

  `control` This is the first modifier for Windows and Linux

  `cmd` This is the first modifier for Mac

  `mod` This is the platform-independent first modifier. It corresponds to »Control« for Windows and Linux, and »Cmd« for Mac.

  `meta`

  `alt`
- You can also select "*none*" if you do not need a modifier.
- You can enter multiple modifiers for a key combination by entering a space-separated list of modifiers, e.g. `alt shift`.

Do not use quotes around the modifiers, enter them in plain text

Name	Data Type	Values	Default
Text	String	–	none

- Use this parameter to enter the text or regular expression (→ page **??**) you want to drag.
- This parameter will be compared to entries in the list, using the operator provided.

Name	Data Type	Values	Default
Operator	String	equals not equals matches simple match	equals

- The *operator* parameter has four possible values.
- "*not equals*" looks for something that does *not exactly* match.
- "*equals*" looks for an *exact* match.
- Select "*simple match*" to use a simple match expression (→ page **??**) .
- Select "*matches*" to use a regular expression (→ page **??**) .
- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

Name	Data Type	Values	Default
Search Type	String	absolute relative	absolute

- The *search type* parameter has two options.
- "*absolute*" begins searching for the value at the first entry.
- Select "*relative*" to begin the search for the value after the currently selected entry. This allows you to search for the next occurence of a value, ignoring all elements before and including the selected entry.

2.2.4.6 Drop on Entry by Index

```
(List Component)
```

- This action drops a list item.
- The item on which to drop the dragged item is specified by its index in the list.

- The cursor is moved over the component.
- The mouse button is let loose, and the item is dropped.
- Precede this action with a *drag* action

Parameters

Name	Data Type	Values	Default
Index	Integer	–	none

- Use this parameter to enter the item you want to drag.
- Use the index of the item to indicate which item to drag.
- For example, if you want to drag the first item from the list, enter: `1`.

Name	Data Type	Values	Default
Delay before drop (milliseconds)	Integer	–	100

- Use this parameter to wait before dropping the item you have dragged.
- This can be useful to give the user interface time to scroll etc.
- Give the time to delay in milliseconds.

2.2.4.7 Drop on Entry by Value

`(List Component)`

- This action drops a list item.
- The item on which to drop the dragged item is specified by its value in the list.
- The cursor is moved over the component.
- The mouse button is let loose, and the item is dropped.
- Precede this action with a *drag* action

Parameters

Name	Data Type	Values	Default
Text	String	–	none

- Use this parameter to enter the text or regular expression (→ page **??**) of theitem you want to drop onto.

- This parameter will be compared to entries in the list, using the operator provided.

Name	Data Type	Values	Default
Operator	String	equals not equals matches simple match	equals

- The *operator* parameter has four possible values.
- "*not equals*" looks for something that does *not exactly* match.
- "*equals*" looks for an *exact* match.
- Select "*simple match*" to use a simple match expression (→ page **??**) .
- Select "*matches*" to use a regular expression (→ page **??**) .
- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

Name	Data Type	Values	Default
Search Type	String	absolute relative	absolute

- The *search type* parameter has two options.
- "*absolute*" begins searching for the value at the first entry.
- Select "*relative*" to begin the search for the value after the currently selected entry. This allows you to search for the next occurence of a value, ignoring all elements before and including the selected entry.

Name	Data Type	Values	Default
Delay before drop (milliseconds)	Integer	–	100

- Use this parameter to wait before dropping the item you have dragged.
- This can be useful to give the user interface time to scroll etc.
- Give the time to delay in milliseconds.

2.2.4.8 Select Entry by Index/Indices

`(List Component)`

- Use this action to select an item from the list.
- Enter the item you want to select using its index.
- This is achieved by clicking on the indicated item

This action is unsupported for the HTML toolkit

Parameters

Name	Data Type	Values	Default
Index	String	–	none

- Use this parameter to enter the item or items you want to select.
- Use numbers separated by commas to indicate which items you want to select. Do not enter a space between list items.
- For example, if you want to select the first, fourth and sixth items from the list, enter: `1,4,6`.

Name	Data Type	Values	Default
Extend Selection	String	yes no	no

- The *Extend Selection* parameter has two possible values.
- "*yes*" indicates the chosen item will be added to the current selection.
- Select "*no*" to begin a new selection with the chosen item.

Name	Data Type	Values	Default
Mouse Button	Integer	1 2 3	1

- Use this parameter to select which mouse button you want to click with.
  - `1` = left mouse button
  - `2` = middle mouse button
  - `3` = right mouse button
- The mouse button you use to click with can also have an effect on what happens.

- The button to click will depend on your AUT and what the click should achieve.

Do not enter quotes around the mouse button numbers, en them in plain text.

Name	Data Type	Values	Default
Number of Clicks	Integer	–	1

- Enter the amount of clicks that should be performed on the component or item.
- If you enter `0`, the mouse pointer will just be moved over the specified component or item.
- Entering `1` (without quotes) will single-click on the component or item.
- Entering `2` (without quotes) will double-click on the component or item.

2.2.4.9 Select Entry by Value(s)

`(List Component)`

- Use this action to select an item or items from the list.
- Enter the items you want to select using their names (values).
- This is achieved by:
  - Pressing and holding »CTRL«
  - Clicking on the indicated items
  - Releasing »CTRL«

This action is unsupported for the HTML toolkit

Parameters

Name	Data Type	Values	Default
Text	String	–	none

- Use this parameter to enter the text or regular expression (→ page **??**) you want to select.

- If you want to select multiple values, separate the values with a comma ', '. Do not enter a space between list items.
- This parameter will be compared to entries in the List, using the Operator provided.

Name	Data Type	Values	Default
Operator	String	equals not equals matches simple match	equals

- The *operator* parameter has four possible values.
- "*not equals*" looks for something that does *not exactly* match.
- "*equals*" looks for an *exact* match.
- Select "*simple match*" to use a simple match expression (→ page **??**) .
- Select "*matches*" to use a regular expression (→ page **??**) .
- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

Name	Data Type	Values	Default
Search Type	String	absolute relative	absolute

- The *search type* parameter has two options.
- "*absolute*" begins searching for the value at the first entry.
- Select "*relative*" to begin the search for the value after the currently selected entry. This allows you to search for the next occurence of a value, ignoring all elements before and including the selected entry.

Name	Data Type	Values	Default
Extend Selection	String	yes no	no

- The *Extend Selection* parameter has two possible values.
- "*yes*" indicates the chosen item will be added to the current selection.
- Select "*no*" to begin a new selection with the chosen item.

Name	Data Type	Values	Default
Mouse Button	Integer	1 2 3	1

- Use this parameter to select which mouse button you want to click with.
  - 1 = left mouse button
  - 2 = middle mouse button
  - 3 = right mouse button
- The mouse button you use to click with can also have an effect on what happens.
- The button to click will depend on your AUT and what the click should achieve.

Do not enter quotes around the mouse button numbers, en them in plain text.

Name	Data Type	Values	Default
Number of Clicks	Integer	–	1

- Enter the amount of clicks that should be performed on the component or item.
- If you enter 0, the mouse pointer will just be moved over the specified component or item.
- Entering 1 (without quotes) will single-click on the component or item.
- Entering 2 (without quotes) will double-click on the component or item.

2.2.5 Menu Bar Component

Description:

- A *menu bar* is the component typically found at the top of an application window.
- It typically contains menus such as "File", "Edit", "Help", etc.

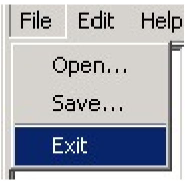


Figure 2.8: *Menu Bar*

Because the forward slash (/) is a special symbol for menus, if you want to use a slash as part of your parameter value, you have to mask it. See the section later in this document (→ page **??**) for more details.

Mapping menu bars

Menu bars do not need to be mapped in the Object Mapping Mode as they are automatically found during test execution.

Actions on menus are not supported in the HTML toolkit.

Synopsis:

- Menu Bar Component (concrete)

New Actions

Name	Type	Parameters
Check Enablement of Entry by Indexpath (→ page 155)	e	String:Indexpath Boolean:Enabled
Check Enablement of Entry by Textpath (→ page 156)	e	String:Menupath String:Operator Boolean:Enabled

Check Existence of Entry by Indexpath (→ page 157)	e	String:Indexpath Boolean:Exists
Check Existence of Entry by Textpath (→ page 158)	e	String:Menupath String:Operator Boolean:Exists
Check Selection of Entry by Indexpath (→ page 159)	e	String:Indexpath Boolean:Selected
Check Selection of Entry by Textpath (→ page 159)	e	String:Menupath String:Operator Boolean:Selected
Select Menu Entry by Indexpath (→ page 161)	e	String:Indexpath
Select Menu Entry by Textpath (→ page 161)	e	String:Menupath String:Operator
Wait for Component (→ page 162)	e	Integer:Timeout in ms Integer:Delay after Visibility

Inherited Actions

none

2.2.5.1 Check Enablement of Entry by Indexpath

`(Menu Bar Component)`

- Use this action to check the enablement status of an item in a menu bar.
- The item to check is given using the indexpath to the item.

Parameters

Name	Data Type	Values	Default
Indexpath	String	–	none

Use this parameter to specify the menupath to the item you want to check.Make sure you give the whole menupath (start from the very beginning of the menu).

- Enter the path to the item as an indexpath.
- Use slash `'/'` as a path separator (to separate parent nodes from child nodes).
- For example, `1/2` (without quotes).

The first node is '1' (without quotes)

Name	Data Type	Values	Default
Enabled	Boolean	true false	true

- Set this parameter to true if you expect the menu item to be enabled.
- Set the parameter to false if you expect the menu item to be disabled.

2.2.5.2 Check Enablement of Entry by Textpath

`(Menu Bar Component)`

- Use this action to check whether an item in a menu is enabled.
- The item to check is given using the textpath to the item.

Parameters

Name	Data Type	Values	Default
Menupath	String	–	none

Use this parameter to specify the menupath to the item you want to check. Make sure you enter the whole menupath.

- Enter the path to the item as a textpath.
- Use slash `'/'` as a path separator (to separate parent nodes from child nodes).
- For example, `File/Open` or `Category/Horror` (without quotes).
- Either make sure that your path is written exactly as it appears in the interface, or use a regular expression to match the text.
- Each segment of the path will be used to find a corresponding node, using the operator provided.

Name	Data Type	Values	Default
Operator	String	equals not equals matches simple match	equals

- The *operator* parameter has four possible values.
- "*not equals*" looks for something that does *not exactly* match.
- "*equals*" looks for an *exact* match.
- Select "*simple match*" to use a simple match expression (→ page **??**) .

- Select "*matches*" to use a regular expression (→ page **??**) .
- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

Name	Data Type	Values	Default
Enabled	Boolean	true false	true

- Set this parameter to true if you expect the menu item to be enabled.
- Set the parameter to false if you expect the menu item to be disabled.

2.2.5.3 Check Existence of Entry by Indexpath

`(Menu Bar Component)`

- Use this action to check the existence of an item in a menu bar.
- The item to check is given using the indexpath to the item.

Parameters

Name	Data Type	Values	Default
Indexpath	String	–	none

Use this parameter to specify the menupath to the item you want to check. Make sure you give the whole menupath (start from the very beginning of the menu).

- Enter the path to the item as an indexpath.
- Use slash `'/'` as a path separator (to separate parent nodes from child nodes).
- For example, `1/2` (without quotes).

The first node is '1' (without quotes)

Name	Data Type	Values	Default
Exists	Boolean	true false	true

- Set this parameter to true if you expect the menu item to exist.
- Set the parameter to false if you expect the menu item to *not* exist.

2.2.5.4 Check Existence of Entry by Textpath

`(Menu Bar Component)`

- Use this action to check the existence of an item in a menu bar.
- The item to check is given using the textpath to the item.

Parameters

Name	Data Type	Values	Default
Menupath	String	–	none

Use this parameter to specify the menupath to the item you want to check. Make sure you enter the whole path.

- Enter the path to the item as a textpath.
- Use slash `'/'` as a path separator (to separate parent nodes from child nodes).
- For example, `File/Open` or `Category/Horror` (without quotes).
- Either make sure that your path is written exactly as it appears in the interface, or use a regular expression to match the text.
- Each segment of the path will be used to find a corresponding node, using the operator provided.

Name	Data Type	Values	Default
Operator	String	equals not equals matches simple match	equals

- The *operator* parameter has four possible values.
- "*not equals*" looks for something that does *not exactly* match.
- "*equals*" looks for an *exact* match.
- Select "*simple match*" to use a simple match expression (→ page **??**) .
- Select "*matches*" to use a regular expression (→ page **??**) .
- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

Name	Data Type	Values	Default
Exists	Boolean	true false	true

- Set this parameter to true if you expect the menu item to exist.
- Set the parameter to false if you expect the menu item to *not* exist.

2.2.5.5 Check Selection of Entry by Indexpath

`(Menu Bar Component)`

- Use this action to check whether an item in a menu is selected.
- You enter the menupath to the item, and whether you expect it to be selected or not.
- The menupath is given as an indexpath.

Parameters

Name	Data Type	Values	Default
Indexpath	String	–	none

Use this parameter to specify the menupath to the item you want to check. Make sure you give the whole menupath (start from the very beginning of the menu).

- Enter the path to the item as an indexpath.
- Use slash `'/'` as a path separator (to separate parent nodes from child nodes).
- For example, `1/2` (without quotes).

The first node is '1' (without quotes)

Name	Data Type	Values	Default
Selected	Boolean	true false	true

- Set this parameter to true if you expect the menu item to be selected.
- Set this parameter to false if you expect the menu item *not* to be selected.

2.2.5.6 Check Selection of Entry by Textpath

`(Menu Bar Component)`

- Use this action to check whether an item in a menu is selected.

- You enter the menupath to the item, and whether you expect it to be selected or not.
- The menupath is given as a textpath.

Parameters

Name	Data Type	Values	Default
Menupath	String	–	none

Use this parameter to specify the menupath to the item you want to check. Make sure you enter the whole path.

- Enter the path to the item as a textpath.
- Use slash `'/'` as a path separator (to separate parent nodes from child nodes).
- For example, `File/Open` or `Category/Horror` (without quotes).
- Either make sure that your path is written exactly as it appears in the interface, or use a regular expression to match the text.
- Each segment of the path will be used to find a corresponding node, using the operator provided.

Name	Data Type	Values	Default
Operator	String	equals not equals matches simple match	equals

- The *operator* parameter has four possible values.
- "*not equals*" looks for something that does *not exactly* match.
- "*equals*" looks for an *exact* match.
- Select "*simple match*" to use a simple match expression (→ page **??**) .
- Select "*matches*" to use a regular expression (→ page **??**) .
- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

Name	Data Type	Values	Default
Selected	Boolean	true false	true

- Set this parameter to true if you expect the menu item to be selected.
- Set this parameter to false if you expect the menu item *not* to be selected.

2.2.5.7 Select Menu Entry by Indexpath

`(Menu Bar Component)`

- Use this action to select an item from a menu bar.
- The item you want to select is given using the indexpath to the item.

Parameters

Name	Data Type	Values	Default
Indexpath	String	–	none

Use this parameter to specify the menupath to the item you want to select.Make sure you give the whole menupath (start from the very beginning of the menu).

- Enter the path to the item as an indexpath.
- Use slash `'/'` as a path separator (to separate parent nodes from child nodes).
- For example, `1/2` (without quotes).

The first node is '1' (without quotes)

2.2.5.8 Select Menu Entry by Textpath

`(Menu Bar Component)`

- Use this action to select an item from a menu bar.
- The item you want to select is given using the textpath to the item.

Parameters

Name	Data Type	Values	Default
Menupath	String	–	none

Use this parameter to specify the menupath to the item you want to select. Make sure you enter the whole path.

- Enter the path to the item as a textpath.
- Use slash `'/'` as a path separator (to separate parent nodes from child nodes).
- For example, `File/Open` or `Category/Horror` (without quotes).
- Either make sure that your path is written exactly as it appears in the interface, or use a regular expression to match the text.

- Each segment of the path will be used to find a corresponding node, using the operator provided.

Name	Data Type	Values	Default
Operator	String	equals not equals matches simple match	equals

- The *operator* parameter has four possible values.
- "*not equals*" looks for something that does *not exactly* match.
- "*equals*" looks for an *exact* match.
- Select "*simple match*" to use a simple match expression (→ page **??**) .
- Select "*matches*" to use a regular expression (→ page **??**) .
- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

2.2.5.9 Wait for Component

`(Menu Bar Component)`

- The wait for menu action lets you wait until the menu is present/loaded before continuing with the next Test Step.

Parameters

Name	Data Type	Values	Default
Timeout in ms	Integer	–	1000

- Enter the amount of time (in milliseconds) to wait before the Test Step fails.
- If the menu does not appear in this time frame, the Test Step is unsuccessful.

Name	Data Type	Values	Default
Delay after Visibility	Integer	–	200

- Sometimes a widget is not ready to receive events (mouse clicks, key presses) until a few moments after becoming visible.
- Use this parameter to set the amount of time (in milliseconds) to wait once the widget has appeared before continuing.

2.2.6 Tab Component

Description:

A *tabbed pane* is a container that allows you to switch between multiple groups of components within a single area.

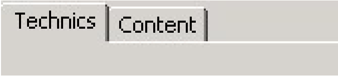


Figure 2.9: *Tabbed Pane*

Mapping tabbed panes

In the Object Mapping Mode, a tabbed pane to be mapped looks like this:

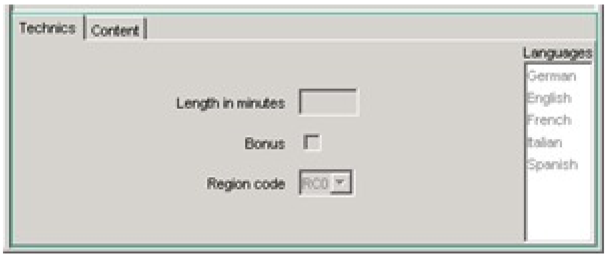


Figure 2.10: *Tabbed Pane*

Synopsis:

- Tab Component (concrete)
  - Graphics Component (abstract)

New Actions

Name	Type	Parameters
Check Enablement of Tab by Index (→ page 169)	c	Integer:Index Boolean:Enabled
Check Enablement of Tab by Value (→ page 170)	c	String:Title String:Operator Boolean:Enabled
Check Existence of Tab (→ page 171)	c	String:Tab String:Operator Boolean:Exists

Check Selection of Tab by Index (→ page 172)	c	Integer:Index Boolean:Selected
Check Selection of Tab by Value (→ page 172)	c	String:Title String:Operator Boolean:Selected
Check Text of Tab by Index (→ page 173)	c	Integer:Index String:Title String:Operator
Select Tab by Index (→ page 174)	e	Integer:Index
Select Tab by Value (→ page 175)	e	String:Title String:Operator

Inherited Actions

Name	Type	Parameters	Inherited from
Check Enablement (→ page 23)	c	Boolean:Enabled	Graphics Component (abstract)
Check Enablement of Context Menu Entry by Indexpath (→ page 24)	c	String:Indexpath Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)
Check Enablement of Context Menu Entry by Indexpath (Specify Position) (→ page 25)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Indexpath Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)
Check Enablement of Context Menu Entry by Textpath (→ page 28)	c	String:Textpath String:Operator Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)
Check Enablement of Context Menu Entry by Textpath (Specify Position) (→ page 29)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Textpath String:Operator Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)
Check Existence (→ page 32)	c	Boolean:Exists	Graphics Component (abstract)
Check Existence of Context Menu Entry by Indexpath (→ page 33)	c	String:Indexpath Boolean:Exists Integer:Mouse Button	Graphics Component (abstract)

Check Existence of Context Menu Entry by Indexpath (Specify Position) (→ page 34)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Indexpath Boolean:Exists Integer:Mouse Button	Graphics Component (abstract)
Check Existence of Context Menu Entry by Textpath (→ page 37)	c	String:Textpath String:Operator Boolean:Exists Integer:Mouse Button	Graphics Component (abstract)
Check Existence of Context Menu Entry by Textpath (Specify Position) (→ page 38)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Textpath String:Operator Boolean:Exists Integer:Mouse Button	Graphics Component (abstract)
Check Focus (→ page 41)	c	Boolean:Has Focus	Graphics Component (abstract)
Check Property (→ page 42)	c	String:Property Name String:Property Value String:Operator	Graphics Component (abstract)
Check Selection of Context Menu Entry by Indexpath (→ page 43)	c	String:Indexpath Boolean:Selected Integer:Mouse Button	Graphics Component (abstract)
Check Selection of Context Menu Entry by Indexpath (Specify Position) (→ page 45)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Indexpath Boolean:Selected Integer:Mouse Button	Graphics Component (abstract)

Check Selection of Context Menu Entry by Textpath (→ page 47)	c	String:Textpath String:Operator Boolean:Selected Integer:Mouse Button	Graphics Component (abstract)
Check Selection of Context Menu Entry by Textpath (Specify Position) (→ page 49)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Textpath String:Operator Boolean:Selected Integer:Mouse Button	Graphics Component (abstract)
Click (→ page 52)	e	Integer:Number of Clicks Integer:Mouse Button	Graphics Component (abstract)
Click in Component (→ page 53)	e	Integer:Number of Clicks Integer:Mouse Button Integer:x-position String:x-units Integer:y-position String:y-units	Graphics Component (abstract)
Drag (→ page 55)	e	Integer:Mouse Button String:Modifier Keys Integer:x-position String:x-units Integer:y-position String:y-units	Graphics Component (abstract)

Drop (→ page 58)	e	Integer:x-position String:x-units Integer:y-position String:y-units Integer:Delay before drop (milliseconds)	Graphics Component (abstract)
Select Context Menu Entry by Indexpath (→ page 60)	e	String:Indexpath Integer:Mouse Button	Graphics Component (abstract)
Select Context Menu Entry by Indexpath (Specify Position) (→ page 61)	e	Integer:x-position String:x-units Integer:y-position String:y-units String:Indexpath Integer:Mouse Button	Graphics Component (abstract)
Select Context Menu Entry by Textpath (→ page 64)	e	String:Textpath String:Operator Integer:Mouse Button	Graphics Component (abstract)
Select Context Menu Entry by Textpath (Specify Position) (→ page 66)	e	Integer:x-position String:x-units Integer:y-position String:y-units String:Menupath String:Operator Integer:Mouse Button	Graphics Component (abstract)

Show Text (→ page 69)	e	String:Text Integer:Text Size (in points) Integer:Time per Word (in milliseconds) Integer:Window Width (in pixels)	Graphics Component (abstract)
Store Property (→ page 70)	e	Variable:Variable Name String:Property Name	Graphics Component (abstract)
Wait for Component (→ page 71)	e	Integer:Timeout in ms Integer:Delay after Visibility	Graphics Component (abstract)

2.2.6.1 Check Enablement of Tab by Index

`(Tab Component)`

- Use this action to check if a tab you specify is currently enabled (selectable/visible).
- You specify the tab using its index.

Parameters

Name	Data Type	Values	Default
Index	Integer	–	none

- Use this parameter to specify which tab you want to check.
- Refer to the tab using its index.
- The first (leftmost) tab has an index of `1`, the second `2` and so on.

Name	Data Type	Values	Default
Enabled	Boolean	true false	true

- Set this parameter to true if you expect the tab whose index you give to be enabled.
- Set the parameter to false if you expect the tab whose index you give to be disabled.

2.2.6.2 Check Enablement of Tab by Value

`(Tab Component)`

- Use this action to check if a tab you specify is currently enabled (selectable/visible).
- You specify the tab using its name.

Parameters

Name	Data Type	Values	Default
Title	String	–	none

- Enter the name of the tab you want to check.
- This parameter will be compared to Tabs in the Pane, using the Operator provided.

Name	Data Type	Values	Default
Operator	String	equals not equals matches simple match	equals

- The *operator* parameter has four possible values.
- "*not equals*" looks for something that does *not exactly* match.
- "*equals*" looks for an *exact* match.
- Select "*simple match*" to use a simple match expression (→ page **??**) .
- Select "*matches*" to use a regular expression (→ page **??**) .
- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

Name	Data Type	Values	Default
Enabled	Boolean	true false	true

- Set this parameter to true if you expect the tab whose name you give to be enabled.
- Set the parameter to false if you expect the tab whose name you give to be disabled.

2.2.6.3 Check Existence of Tab

(Tab Component)

- Use this action to check whether a tab in a tabbed pane exists.
- The tab can be checked based on its title or its index.

Parameters

Name	Data Type	Values	Default
Tab	String	–	none

- Enter the title (e.g. `DVDs`) or index (e.g. `1`) of the tab whose existence you want to check.

Name	Data Type	Values	Default
Operator	String	equals not equals matches simple match	equals

- The *operator* parameter has four possible values.
- "*not equals*" looks for something that does *not exactly* match.
- "*equals*" looks for an *exact* match.
- Select "*simple match*" to use a simple match expression (→ page **??**) .
- Select "*matches*" to use a regular expression (→ page **??**) .
- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

Name	Data Type	Values	Default
Exists	Boolean	true false	true

- Set this parameter to true if you expect the tab to exist.
- Set this parameter to false if you expect the tab not to exist.

2.2.6.4 Check Selection of Tab by Index

`(Tab Component)`

- Use this action to check if a tab you specify is currently enabled (selectable/visible).
- You specify the tab using its index.

Parameters

Name	Data Type	Values	Default
Index	Integer	–	none

- Use this parameter to specify which tab you want to check.
- Refer to the tab using its index.
- The first (leftmost) tab has an index of `1`, the second `2` and so on.

Name	Data Type	Values	Default
Selected	Boolean	true false	true

- Set this parameter to true if you expect the tab whose index you give to be enabled.
- Set the parameter to false if you expect the tab whose index you give to be disabled.

2.2.6.5 Check Selection of Tab by Value

`(Tab Component)`

- Use this action to check if the tab you specify is currently selected/activated.
- You specify the tab using its title.

Parameters

Name	Data Type	Values	Default
Title	String	–	none

- Enter the name of the tab you want to check.
- This parameter will be compared to entries in the Combo Box, using the Operator provided.

Name	Data Type	Values	Default
Operator	String	equals not equals matches simple match	equals

- The *operator* parameter has four possible values.
- "*not equals*" looks for something that does *not exactly* match.
- "*equals*" looks for an *exact* match.
- Select "*simple match*" to use a simple match expression (→ page **??**) .
- Select "*matches*" to use a regular expression (→ page **??**) .
- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

Name	Data Type	Values	Default
Selected	Boolean	true false	true

- Set this parameter to true if you expect the tab to be selected.
- Set this parameter to false if you expect the tab not to be selected.

2.2.6.6 Check Text of Tab by Index

`(Tab Component)`

- Use this action to check the text of a tab in a tabbed pane.
- You enter the index of the tab you want to check and the text you expect.

Parameters

Name	Data Type	Values	Default
Index	Integer	–	none

- Enter the index of the tab whose text you want to check.

Name	Data Type	Values	Default
Title	String	–	none

- Enter the title of the tab that you want to check at this index.

Name	Data Type	Values	Default
Operator	String	equals not equals matches simple match	equals

- The *operator* parameter has four possible values.
- "*not equals*" looks for something that does *not exactly* match.
- "*equals*" looks for an *exact* match.
- Select "*simple match*" to use a simple match expression (→ page **??**) .
- Select "*matches*" to use a regular expression (→ page **??**) .
- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

2.2.6.7 Select Tab by Index

`(Tab Component)`

- Use this action to select/activate a tab.
- You choose the tab to select by giving its index.
- This is achieved by clicking on the tab whose index you enter.
- In RCP AUT's, the index of the tab refers to the order in which the tabs were opened. So the first tab to be opened has an index of `1`. This action selects the tab even if the tab with the given index is currently not visible (e.g. in the editor view).

Parameters

Name	Data Type	Values	Default
Index	Integer	–	none

- Use this parameter to specify which tab you want to select/activate.
- Refer to the tab using its index.
- The first (leftmost) tab has an index of `1`, the second `2` and so on.

2.2.6.8 Select Tab by Value

`(Tab Component)`

- Use this action to select/activate a tab.
- You choose the tab to select by giving its title.
- This is achieved by clicking on the tab whose title you enter.
- In RCP AUT's, this will also select tabs which are not currently visible (e.g. in the editor area).
- If more than one tab name corresponds to the value, the the tab with this value which was opened first will be selected.

Parameters

Name	Data Type	Values	Default
Title	String	–	none

- Use this parameter to specify the tab you want to select.
- This parameter will be compared to Tabs in the Pane, using the Operator provided.

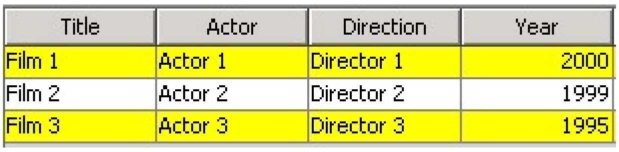
Name	Data Type	Values	Default
Operator	String	equals not equals matches simple match	equals

- The *operator* parameter has four possible values.
- "*not equals*" looks for something that does *not exactly* match.
- "*equals*" looks for an *exact* match.
- Select "*simple match*" to use a simple match expression (→ page **??**) .
- Select "*matches*" to use a regular expression (→ page **??**) .
- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

2.2.7 Table Component

Description:

- A *table* is a component in which data is displayed and edited.
- The format of a table is a two-dimensional layout of cells which are organized into columns and rows.

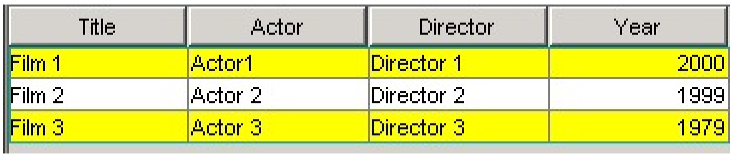


Title	Actor	Direction	Year
Film 1	Actor 1	Director 1	2000
Film 2	Actor 2	Director 2	1999
Film 3	Actor 3	Director 3	1995

Figure 2.11: *Table*

Mapping tables

In the Object Mapping Mode, a table to be mapped looks like this:



Title	Actor	Director	Year
Film 1	Actor1	Director 1	2000
Film 2	Actor 2	Director 2	1999
Film 3	Actor 3	Director 3	1979

Figure 2.12: *Table*

Virtual tables in SWT

The current support for tables has not been verified for virtual SWT tables.

Synopsis:

- Table Component (concrete)
  - Text Input Component (abstract)
    - Text Component (abstract)
      - Graphics Component (abstract)

New Actions

Name	Type	Parameters
Check Editability (Specify Cell) (→ page 185)	c	Boolean:Editable String:Row String:Row Operator String:Column String:Column Operator
Check Editability of Cell (Mouse Position) (→ page 187)	c	Boolean:Editable
Check Editability of Selected Cell (→ page 188)	c	Boolean:Editable
Check Existence of Value in Column (→ page 188)	c	String:Column String:Column Operator String:Value String:Value Operator String:Search Type Boolean:Exists
Check Existence of Value in Row (→ page 190)	c	String:Row String:Row Operator String:Value String:Value Operator String:Search Type Boolean:Exists
Check Selection of Checkbox at Mouse Position (→ page 192)	c	Boolean:Checked
Check Selection of Checkbox in Selected Row (→ page 193)	c	Boolean:Checked
Check Text (Mouse Position) (→ page 193)	c	String:Text String:Operator
Check Text (Specify Cell) (→ page 194)	c	String:Text String:Text Operator String:Row String:Row Operator String:Column String:Column Operator

Drag Cell (→ page 196)	e	Integer:Mouse Button String:Modifier Keys String:Row String:Row Operator String:Column String:Column Operator Integer:x-position String:x-units Integer:y-position String:y-units
Drag Cell from Column (→ page 201)	e	Integer:Mouse Button String:Modifier Keys String:Column String:Column Operator String:Value String:Value Operator String:Search Type
Drag Cell from Row (→ page 204)	e	Integer:Mouse Button String:Modifier Keys String:Row String:Row Operator String:Value String:Value Operator String:Search Type
Drop on Cell (→ page 207)	e	String:Row String:Row Operator String:Column String:Column Operator Integer:x-position String:x-units Integer:y-position String:y-units Integer:Delay before drop (milliseconds)
Drop on Cell from Column (→ page 210)	e	String:Column String:Column Operator String:Value String:Value Operator String:Search Type Integer:Delay before drop (milliseconds)

Drop on Cell from Row (→ page 212)	e	String:Row String:Row Operator String:Value String:Value Operator String:Search Type Integer:Delay before drop (milliseconds)
Input Text (Specify Cell) (→ page 214)	e	String:Text String:Row String:Row Operator String:Column String:Column Operator
Move (→ page 216)	e	String:Direction Integer:Number of Cells Integer:Number of Clicks Integer:x-position String:x-units Integer:y-position String:y-units String:Extend Selection
Replace Text (Specify Cell) (→ page 219)	e	String:Text String:Row String:Row Operator String:Column String:Column Operator
Select Cell (→ page 221)	e	String:Row String:Row Operator String:Column String:Column Operator Integer:Number of Clicks Integer:x-position String:x-units Integer:y-position String:y-units String:Extend Selection Integer:Mouse Button

Select Value from Column (→ page 225)	e	String:Column String:Column Operator String:Value String:Value Operator Integer:Number of Clicks String:Extend Selection String:Search Type Integer:Mouse Button
Select Value from Row (→ page 228)	e	String:Row String:Row Operator String:Value String:Value Operator Integer:Number of Clicks String:Extend Selection String:Search Type Integer:Mouse Button
Store Value (Specify Cell) (→ page 231)	e	Variable:Variable Name String:Row String:Row Operator String:Column String:Column Operator
Store Value at Mouse Position (→ page 233)	e	Variable:Variable Name
Toggle Checkbox at Mouse Position (→ page 234)	e	
Toggle Checkbox in Selected Row (→ page 234)	e	

Inherited Actions

Name	Type	Parameters	Inherited from
Check Editability (→ page 84)	c	Boolean:Editable	Text Input Component (abstract)
Check Enablement (→ page 23)	c	Boolean:Enabled	Graphics Component (abstract)
Check Enablement of Context Menu Entry by Indexpath (→ page 24)	c	String:Indexpath Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)
Check Enablement of Context Menu Entry by Indexpath (Specify Position) (→ page 25)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Indexpath Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)
Check Enablement of Context Menu Entry by Textpath (→ page 28)	c	String:Textpath String:Operator Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)
Check Enablement of Context Menu Entry by Textpath (Specify Position) (→ page 29)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Textpath String:Operator Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)

Check Existence (→ page 32)	c	Boolean:Exists	Graphics Component (abstract)
Check Existence of Context Menu Entry by Indexpath (→ page 33)	c	String:Indexpath Boolean:Exists Integer:Mouse Button	Graphics Component (abstract)
Check Existence of Context Menu Entry by Indexpath (Specify Position) (→ page 34)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Indexpath Boolean:Exists Integer:Mouse Button	Graphics Component (abstract)
Check Existence of Context Menu Entry by Textpath (→ page 37)	c	String:Textpath String:Operator Boolean:Exists Integer:Mouse Button	Graphics Component (abstract)
Check Existence of Context Menu Entry by Textpath (Specify Position) (→ page 38)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Textpath String:Operator Boolean:Exists Integer:Mouse Button	Graphics Component (abstract)
Check Focus (→ page 41)	c	Boolean:Has Focus	Graphics Component (abstract)
Check Property (→ page 42)	c	String:Property Name String:Property Value String:Operator	Graphics Component (abstract)

Check Selection of Context Menu Entry by Indexpath (→ page 43)	c	String:Indexpath Boolean:Selected Integer:Mouse Button	Graphics Component (abstract)
Check Selection of Context Menu Entry by Indexpath (Specify Position) (→ page 45)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Indexpath Boolean:Selected Integer:Mouse Button	Graphics Component (abstract)
Check Selection of Context Menu Entry by Textpath (→ page 47)	c	String:Textpath String:Operator Boolean:Selected Integer:Mouse Button	Graphics Component (abstract)
Check Selection of Context Menu Entry by Textpath (Specify Position) (→ page 49)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Textpath String:Operator Boolean:Selected Integer:Mouse Button	Graphics Component (abstract)
Check Text (→ page 76)	c	String:Text String:Operator	Text Component (abstract)
Click (→ page 52)	e	Integer:Number of Clicks Integer:Mouse Button	Graphics Component (abstract)
Click in Component (→ page 53)	e	Integer:Number of Clicks Integer:Mouse Button Integer:x-position String:x-units Integer:y-position String:y-units	Graphics Component (abstract)

Drag (→ page 55)	e	Integer:Mouse Button String:Modifier Keys Integer:x-position String:x-units Integer:y-position String:y-units	Graphics Component (abstract)
Drop (→ page 58)	e	Integer:x-position String:x-units Integer:y-position String:y-units Integer:Delay before drop (milliseconds)	Graphics Component (abstract)
Input Text (→ page 84)	e	String:Text	Text Input Component (abstract)
Replace Text (→ page 85)	e	String:Text	Text Input Component (abstract)
Select Context Menu Entry by Indexpath (→ page 60)	e	String:Indexpath Integer:Mouse Button	Graphics Component (abstract)
Select Context Menu Entry by Indexpath (Specify Position) (→ page 61)	e	Integer:x-position String:x-units Integer:y-position String:y-units String:Indexpath Integer:Mouse Button	Graphics Component (abstract)

Select Context Menu Entry by Textpath (→ page 64)	e	String:Textpath String:Operator Integer:Mouse Button	Graphics Component (abstract)
Select Context Menu Entry by Textpath (Specify Position) (→ page 66)	e	Integer:x-position String:x-units Integer:y-position String:y-units String:Menupath String:Operator Integer:Mouse Button	Graphics Component (abstract)
Show Text (→ page 69)	e	String:Text Integer:Text Size (in points) Integer:Time per Word (in milliseconds) Integer:Window Width (in pixels)	Graphics Component (abstract)
Store Property (→ page 70)	e	Variable:Variable Name String:Property Name	Graphics Component (abstract)
Store Value (→ page 77)	e	Variable:Variable Name	Text Component (abstract)
Wait for Component (→ page 71)	e	Integer:Timeout in ms Integer:Delay after Visibility	Graphics Component (abstract)

2.2.7.1 Check Editability (Specify Cell)

`(Table Component)`

- Use this action to check if a cell in a table is editable.
- You enter the cell you want to check and whether you expect it to be editable or not.

This action is unsupported for the HTML toolkit

Parameters

Name	Data Type	Values	Default
Editable	Boolean	true false	false

- Set the parameter to true if you expect the cell to be editable.
- Set the parameter to false if you expect the cell *not* to be editable.

Name	Data Type	Values	Default
Row	String	–	none

Use this parameter to specify the row which contains the cell you want to check.

- In tables, you can address the row of a table either by using its index or by entering the value in the first column of the particular row.
- The first row is `1` (without quotes) and so on.
- The row of table headers at the top of the table can be addressed using the index `0` (without quotes).
- If the cell you want to execute an action on is in the fourth row and contains the value *Price* in the first column, you can address the row either with the index `4` (without quotes) or with the string *Price*.

Name	Data Type	Values	Default
Row Operator	String	equals not equals matches simple match	equals

Enter the operator you want to use for the selection of the row when you enter the row as a string.

- The *operator* parameter has four possible values.
- "*not equals*" looks for something that does *not exactly* match.
- "*equals*" looks for an *exact* match.
- Select "*simple match*" to use a simple match expression (→ page **??**) .
- Select "*matches*" to use a regular expression (→ page **??**) .
- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

Name	Data Type	Values	Default
Column	String	–	none

Use this parameter to enter the column number or header title which contains the cell you want to check.

- In tables, you can address the header of a table either by using its index or by entering its title.
- The first column is `1` (without quotes) and so on. The index `0` (without quotes) refers to the header itself.
- If the cell you want to execute an action on is in the fourth column and is called *Price*, you can address the column either with the index `4` (without quotes) or with the string *Price*.
- In JavaFX AUT's, you can also enter a path to the column to test tables that have nested columns (i.e. where a column has child columns). Enter the column as a path, separated with a forward slash, e.g. `Address/Street`. You can use indexpaths and textpaths for columns in JavaFX tables.

Name	Data Type	Values	Default
Column Operator	String	equals not equals matches simple match	equals

Enter the operator you want to use for the selection of the column when you enter the column as a string.

- The *operator* parameter has four possible values.
- "*not equals*" looks for something that does *not exactly* match.
- "*equals*" looks for an *exact* match.
- Select "*simple match*" to use a simple match expression (→ page **??**) .
- Select "*matches*" to use a regular expression (→ page **??**) .
- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

2.2.7.2 Check Editability of Cell (Mouse Position)

`(Table Component)`

- Use this action to check whether a table cell at the current mouse position is editable or not.

This action is unsupported for the HTML toolkit

Parameters

Name	Data Type	Values	Default
Editable	Boolean	true false	false

- Set this parameter to true if you expect the table cell to be editable.
- Set this parameter to false if you expect the table cell *not* to be editable.

2.2.7.3 Check Editability of Selected Cell

`(Table Component)`

- Use this action to check whether the currently selected table cell is editable or not.

This action is unsupported for the HTML toolkit

Parameters

Name	Data Type	Values	Default
Editable	Boolean	true false	false

- Set this parameter to true if you expect the table cell to be editable.
- Set this parameter to false if you expect the table cell *not* to be editable.

2.2.7.4 Check Existence of Value in Column

`(Table Component)`

- Use this action to check whether a specific value exists in a column you specify.

Parameters

Name	Data Type	Values	Default
Column	String	–	none

- In tables, you can address the header of a table either by using its index or by entering its title.
- The first column is `1` (without quotes) and so on. The index `0` (without quotes) refers to the header itself.
- If the cell you want to execute an action on is in the fourth column and is called *Price*, you can address the column either with the index `4` (without quotes) or with the string *Price*.
- In JavaFX AUT's, you can also enter a path to the column to test tables that have nested columns (i.e. where a column has child columns). Enter the column as a path, separated with a forward slash, e.g. `Address/Street`. You can use indexpaths and textpaths for columns in JavaFX tables.

Name	Data Type	Values	Default
Column Operator	String	equals not equals matches simple match	equals

Enter the operator you want to use for the selection of the column when you enter the column as a string.

- The *operator* parameter has four possible values.
- "*not equals*" looks for something that does *not exactly* match.
- "*equals*" looks for an *exact* match.
- Select "*simple match*" to use a simple match expression (→ page **??**) .
- Select "*matches*" to use a regular expression (→ page **??**) .
- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

Name	Data Type	Values	Default
Value	String	–	none

- Enter the value whose existence you want to check.

Name	Data Type	Values	Default
Value Operator	String	equals not equals matches simple match	equals

Enter the operator for the value you want to check.

- The *operator* parameter has four possible values.
- "*not equals*" looks for something that does *not exactly* match.
- "*equals*" looks for an *exact* match.
- Select "*simple match*" to use a simple match expression (→ page **??**) .
- Select "*matches*" to use a regular expression (→ page **??**) .
- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

Name	Data Type	Values	Default
Search Type	String	absolute relative	absolute

- The *search type* parameter has two options.
- "*absolute*" begins searching for the value at the first entry.
- Select "*relative*" to begin the search for the value after the currently selected entry. This allows you to search for the next occurence of a value, ignoring all elements before and including the selected entry.

Name	Data Type	Values	Default
Exists	Boolean	true false	true

- Set this parameter to true if you expect the value to exist.
- Set this parameter to false if you expect the value *not* to exist.

2.2.7.5 Check Existence of Value in Row

`(Table Component)`

- Use this action to check whether a specific value exists in a row you specify.

Parameters

Name	Data Type	Values	Default
Row	String	–	none

- In tables, you can address the row of a table either by using its index or by entering the value in the first column of the particular row.

- The first row is `1` (without quotes) and so on.
- The row of table headers at the top of the table can be addressed using the index `0` (without quotes).
- If the cell you want to execute an action on is in the fourth row and contains the value *Price* in the first column, you can address the row either with the index `4` (without quotes) or with the string *Price*.

Name	Data Type	Values	Default
Row Operator	String	equals not equals matches simple match	equals

Enter the operator you want to use for the selection of the row when you enter the row as a string.

- The *operator* parameter has four possible values.
- "*not equals*" looks for something that does *not exactly* match.
- "*equals*" looks for an *exact* match.
- Select "*simple match*" to use a simple match expression (→ page **??**) .
- Select "*matches*" to use a regular expression (→ page **??**) .
- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

Name	Data Type	Values	Default
Value	String	–	none

- Enter the value whose existence you want to check.

Name	Data Type	Values	Default
Value Operator	String	equals not equals matches simple match	equals

Enter the operator for the value you want to check.

- The *operator* parameter has four possible values.
- "*not equals*" looks for something that does *not exactly* match.
- "*equals*" looks for an *exact* match.
- Select "*simple match*" to use a simple match expression (→ page **??**) .
- Select "*matches*" to use a regular expression (→ page **??**) .

- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

Name	Data Type	Values	Default
Search Type	String	absolute relative	absolute

- The *search type* parameter has two options.
- "*absolute*" begins searching for the value at the first entry.
- Select "*relative*" to begin the search for the value after the currently selected entry. This allows you to search for the next occurence of a value, ignoring all elements before and including the selected entry.

Name	Data Type	Values	Default
Exists	Boolean	true false	true

- Set this parameter to true if you expect the value to exist.
- Set this parameter to false if you expect the value *not* to exist.

2.2.7.6 Check Selection of Checkbox at Mouse Position

`(Table Component)`

- This action checks whether a checkbox in the first cell of a table is selected.
- To use this action, the current mouse position must be over the correct cell. You can achieve this using the select actions, either with 0 clicks or with 1 click.
- This action is currently only supported for SWT/RCP AUT's.

Parameters

Name	Data Type	Values	Default
Checked	Boolean	true false	true

- Set to true if you expect the checkbox to be selected.
- Set to false if you expect the checkbox *not* to be selected.

2.2.7.7 Check Selection of Checkbox in Selected Row

`(Table Component)`

- This action checks whether a checkbox in the first cell of a table is selected.
- To use this action, the row must already be selected.
- This action is currently only supported for SWT/RCP AUT's.

Parameters

Name	Data Type	Values	Default
Checked	Boolean	true false	true

- Set to true if you expect the checkbox to be selected.
- Set to false if you expect the checkbox *not* to be selected.

2.2.7.8 Check Text (Mouse Position)

`(Table Component)`

- Use this action to check the text in a cell in a table based on the current position of the cursor.
- This can be used in conjunction with the action "*select cell*" with a click count of 0.

This action is unsupported for the HTML toolkit

Parameters

Name	Data Type	Values	Default
Text	String	–	none

- Enter the text you want to check.
- This parameter will be compared to the text in the cell, using the operator provided.

Name	Data Type	Values	Default
Operator	String	equals not equals matches simple match	equals

- The *operator* parameter has four possible values.
- "*not equals*" looks for something that does *not exactly* match.
- "*equals*" looks for an *exact* match.
- Select "*simple match*" to use a simple match expression (→ page **??**) .
- Select "*matches*" to use a regular expression (→ page **??**) .
- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

2.2.7.9 Check Text (Specify Cell)

`(Table Component)`

- Use this action to check whether the text in a cell matches a given value.
- You specify the text you want to check and the row and column values for the cell.

Parameters

Name	Data Type	Values	Default
Text	String	–	none

- Enter the text you want to check against the text in the cell.
- This parameter will be compared to the text in the cell, using the Operator provided.

Name	Data Type	Values	Default
Text Operator	String	equals not equals matches simple match	equals

Use this parameter to specify the operator used for the cell text.

- The *operator* parameter has four possible values.
- "*not equals*" looks for something that does *not exactly* match.

- "*equals*" looks for an *exact* match.
- Select "*simple match*" to use a simple match expression (→ page **??**) .
- Select "*matches*" to use a regular expression (→ page **??**) .
- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

Name	Data Type	Values	Default
Row	String	–	none

Use this parameter to specify the row which contains the cell whose text you want to check.

- In tables, you can address the row of a table either by using its index or by entering the value in the first column of the particular row.
- The first row is `1` (without quotes) and so on.
- The row of table headers at the top of the table can be addressed using the index `0` (without quotes).
- If the cell you want to execute an action on is in the fourth row and contains the value *Price* in the first column, you can address the row either with the index `4` (without quotes) or with the string *Price*.

Name	Data Type	Values	Default
Row Operator	String	equals not equals matches simple match	equals

Enter the operator you want to use for the selection of the row when you enter the row as a string.

- The *operator* parameter has four possible values.
- "*not equals*" looks for something that does *not exactly* match.
- "*equals*" looks for an *exact* match.
- Select "*simple match*" to use a simple match expression (→ page **??**) .
- Select "*matches*" to use a regular expression (→ page **??**) .
- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

Name	Data Type	Values	Default
Column	String	–	none

Use this parameter to specify the column which contains the cell whose text you want to check.

- In tables, you can address the header of a table either by using its index or by entering its title.
- The first column is `1` (without quotes) and so on. The index `0` (without quotes) refers to the header itself.
- If the cell you want to execute an action on is in the fourth column and is called *Price*, you can address the column either with the index `4` (without quotes) or with the string *Price*.
- In JavaFX AUT's, you can also enter a path to the column to test tables that have nested columns (i.e. where a column has child columns). Enter the column as a path, separated with a forward slash, e.g. `Address/Street`. You can use indexpaths and textpaths for columns in JavaFX tables.

Name	Data Type	Values	Default
Column Operator	String	equals not equals matches simple match	equals

Enter the operator you want to use for the selection of the column when you enter the column as a string.

- The *operator* parameter has four possible values.
- "*not equals*" looks for something that does *not exactly* match.
- "*equals*" looks for an *exact* match.
- Select "*simple match*" to use a simple match expression (→ page **??**) .
- Select "*matches*" to use a regular expression (→ page **??**) .
- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

2.2.7.10 Drag Cell

`(Table Component)`

- This action drags the table cell you specify.
- The cell is specified by its row and column values.
- The cursor is moved over the cell.
- The mouse button you specify is held, and the cell is dragged.
- You **must** follow this action with a *drop* action

If you want to drag multiple items, use an action to select the items you want to drag before using the drag action.

This action is unsupported for the HTML toolkit

Parameters

Name	Data Type	Values	Default
Mouse Button	Integer	1 2 3	1

- Use this parameter to select which mouse button you want to click with.
  - `1` = left mouse button
  - `2` = middle mouse button
  - `3` = right mouse button
- The mouse button you use to click with can also have an effect on what happens.
- The button to click will depend on your AUT and what the click should achieve.

Do not enter quotes around the mouse button numbers, en them in plain text.

Name	Data Type	Values	Default
Modifier Keys	String	none shift control alt meta cmd mod	none

- Use this parameter to specify a key or keys to be pressed at the same time as the action is being executed.
- You can use this parameter to specify the following keys, or combinations thereof:

`shift`

`control` This is the first modifier for Windows and Linux

`cmd` This is the first modifier for Mac

`mod` This is the platform-independent first modifier. It corresponds to »CONTROL« for Windows and Linux, and »CMD« for Mac.

`meta`

`alt`

- You can also select *"none"* if you do not need a modifier.
- You can enter multiple modifiers for a key combination by entering a space-separated list of modifiers, e.g. `alt shift`.

Do not use quotes around the modifiers, enter them in plain text

Name	Data Type	Values	Default
Row	String	–	none

Use this parameter to specify the row containing the cell you want to drag.

- In tables, you can address the row of a table either by using its index or by entering the value in the first column of the particular row.
- The first row is `1` (without quotes) and so on.
- The row of table headers at the top of the table can be addressed using the index `0` (without quotes).
- If the cell you want to execute an action on is in the fourth row and contains the value *Price* in the first column, you can address the row either with the index `4` (without quotes) or with the string *Price*.

Name	Data Type	Values	Default
Row Operator	String	equals not equals matches simple match	equals

Enter the operator you want to use for the selection of the row when you enter the row as a string.

- The *operator* parameter has four possible values.
- *"not equals"* looks for something that does *not exactly* match.
- *"equals"* looks for an *exact* match.
- Select *"simple match"* to use a simple match expression (→ page **??**) .
- Select *"matches"* to use a regular expression (→ page **??**) .
- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

Name	Data Type	Values	Default
Column	String	–	none

Use this parameter to specify the column which contains the cell you want to drag.

- In tables, you can address the header of a table either by using its index or by entering its title.
- The first column is `1` (without quotes) and so on. The index `0` (without quotes) refers to the header itself.
- If the cell you want to execute an action on is in the fourth column and is called *Price*, you can address the column either with the index `4` (without quotes) or with the string *Price*.
- In JavaFX AUT's, you can also enter a path to the column to test tables that have nested columns (i.e. where a column has child columns). Enter the column as a path, separated with a forward slash, e.g. `Address/Street`. You can use indexpaths and textpaths for columns in JavaFX tables.

Name	Data Type	Values	Default
Column Operator	String	equals not equals matches simple match	equals

Enter the operator you want to use for the selection of the column when you enter the column as a string.

- The *operator* parameter has four possible values.
- "*not equals*" looks for something that does *not exactly* match.
- "*equals*" looks for an *exact* match.
- Select "*simple match*" to use a simple match expression (→ page **??**) .
- Select "*matches*" to use a regular expression (→ page **??**) .
- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

Name	Data Type	Values	Default
x-position	Integer	–	50

- With this parameter, you can specify where in the component you should click.
- Use this parameter to enter a value for the x-axis (the horizontal position).
- Using the x-units parameter, you can specify whether the value should be in pixels or in percent.
- For example, if you enter `50` and then `percent` (without quotes) , then the click will be performed in the middle of the x-axis.
- All positions are calculated from the upper left of the component, with the positive x-axis stretching toward the right side of the component.

We recommend not using `0` (pixels and percent) or `100`(percent) positions.

Name	Data Type	Values	Default
x-units	String	percent pixel	percent

- Use this parameter to specify whether you want your values for the x-coordinate to be in pixels or percent.
- If you choose percent for this parameter, and `50` for the *x-position* parameter, the click will always be in the middle of the x-axis.

Name	Data Type	Values	Default
y-position	Integer	–	50

- With this parameter, you can specify where in the component you should click.
- Use this parameter to enter a value for the y-axis (the vertical position).
- Using the y-units parameter, you can specify whether the value should be in pixels or in percent.
- For example, if you enter `50` and then `percent` (without quotes), then the click will be performed in the middle of the y-axis.
- All positions are calculated from the upper left of the component, with the positive y-axis stretching toward the bottom of the component.

We recommend not using `0` (pixels and percent) or `100`(percent) as positions.

Name	Data Type	Values	Default
y-units	String	percent pixel	percent

- With this parameter, you can specify where in the component you should click.
- Use this parameter to enter a value for the x-axis (the horizontal position).
- Using the x-units parameter, you can specify whether the value should be in pixels or in percent.
- For example, if you enter `50` and then `percent` (without quotes) , then the click will be performed in the middle of the x-axis.
- All positions are calculated from the upper left of the component, with the positive x-axis stretching toward the right side of the component.

We recommend not using `0` (pixels and percent) or `100`(percent) as positions.

2.2.7.11 Drag Cell from Column

`(Table Component)`

- This action drags the table cell you specify.
- The cell is specified by its column and the value in the cell.
- The cursor is moved over the cell.
- The mouse button you specify is held, and the cell is dragged.
- You **muxt** follow this action with a *drop* action.

If you want to drag multiple items, use an action to select the items you want to drag before using the drag action.

This action is unsupported for the HTML toolkit

Parameters

Name	Data Type	Values	Default
Mouse Button	Integer	1 2 3	1

- Use this parameter to select which mouse button you want to click with.
  - `1` = left mouse button
  - `2` = middle mouse button
  - `3` = right mouse button
- The mouse button you use to click with can also have an effect on what happens.
- The button to click will depend on your AUT and what the click should achieve.

Do not enter quotes around the mouse button numbers, en them in plain text.

Name	Data Type	Values	Default
Modifier Keys	String	none shift control alt meta cmd mod	none

- Use this parameter to specify a key or keys to be pressed at the same time as the action is being executed.
- You can use this parameter to specify the following keys, or combinations thereof:

  shift

  control This is the first modifier for Windows and Linux

  cmd This is the first modifier for Mac

  mod This is the platform-independent first modifier. It corresponds to »CONTROL« for Windows and Linux, and »CMD« for Mac.

  meta

  alt
- You can also select "*none*" if you do not need a modifier.
- You can enter multiple modifiers for a key combination by entering a space-separated list of modifiers, e.g. `alt shift`.

Do not use quotes around the modifiers, enter them in plain text

Name	Data Type	Values	Default
Column	String	–	none

Use this parameter to specify the column which contains the cell value you want to drag.

- In tables, you can address the header of a table either by using its index or by entering its title.
- The first column is `1` (without quotes) and so on. The index `0` (without quotes) refers to the header itself.
- If the cell you want to execute an action on is in the fourth column and is called *Price*, you can address the column either with the index `4` (without quotes) or with the string *Price*.
- In JavaFX AUT's, you can also enter a path to the column to test tables that have nested columns (i.e. where a column has child columns). Enter the column as a path, separated with a forward slash, e.g. `Address/Street`. You can use indexpaths and textpaths for columns in JavaFX tables.

Name	Data Type	Values	Default
Column Operator	String	equals not equals matches simple match	equals

Enter the operator you want to use for the selection of the column when you enter the column as a string.

- The *operator* parameter has four possible values.
- "*not equals*" looks for something that does *not exactly* match.

- "*equals*" looks for an *exact* match.
- Select "*simple match*" to use a simple match expression (→ page **??**) .
- Select "*matches*" to use a regular expression (→ page **??**) .
- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

Name	Data Type	Values	Default
Value	String	–	none

- Enter the value which is in the cell you want to drag.
- This parameter will be compared to cells in the column, using the operator provided.

Name	Data Type	Values	Default
Value Operator	String	equals not equals matches simple match	equals

Use this parameter to specify the operator used for the cell value.

- The *operator* parameter has four possible values.
- "*not equals*" looks for something that does *not exactly* match.
- "*equals*" looks for an *exact* match.
- Select "*simple match*" to use a simple match expression (→ page **??**) .
- Select "*matches*" to use a regular expression (→ page **??**) .
- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

Name	Data Type	Values	Default
Search Type	String	absolute relative	absolute

- The *search type* parameter has two options.
- "*absolute*" begins searching for the value at the first entry.
- Select "*relative*" to begin the search for the value after the currently selected entry. This allows you to search for the next occurence of a value, ignoring all elements before and including the selected entry.

2.2.7.12 Drag Cell from Row

`(Table Component)`

- This action drags the table cell you specify.
- The cell is specified by its row and the value in the cell.
- The cursor is moved over the cell.
- The mouse button you specify is held, and the cell is dragged.
- You **must** follow this action with a *drop* action.

If you want to drag multiple items, use an action to select the items you want to drag before using the drag action.

This action is unsupported for the HTML toolkit

Parameters

Name	Data Type	Values	Default
Mouse Button	Integer	1 2 3	1

- Use this parameter to select which mouse button you want to click with.
  - `1` = left mouse button
  - `2` = middle mouse button
  - `3` = right mouse button
- The mouse button you use to click with can also have an effect on what happens.
- The button to click will depend on your AUT and what the click should achieve.

Do not enter quotes around the mouse button numbers, enter them in plain text.

Name	Data Type	Values	Default
Modifier Keys	String	none shift control alt meta cmd mod	none

- Use this parameter to specify a key or keys to be pressed at the same time as the action is being executed.
- You can use this parameter to specify the following keys, or combinations thereof:

  `shift`

  `control` This is the first modifier for Windows and Linux

  `cmd` This is the first modifier for Mac

  `mod` This is the platform-independent first modifier. It corresponds to »Control« for Windows and Linux, and »Cmd« for Mac.

  `meta`

  `alt`
- You can also select *"none"* if you do not need a modifier.
- You can enter multiple modifiers for a key combination by entering a space-separated list of modifiers, e.g. `alt shift`.

Do not use quotes around the modifiers, enter them in plain te

Name	Data Type	Values	Default
Row	String	–	none

Use this parameter to specify the row which contains the cell you want to drag.

- In tables, you can address the row of a table either by using its index or by entering the value in the first column of the particular row.
- The first row is `1` (without quotes) and so on.
- The row of table headers at the top of the table can be addressed using the index `0` (without quotes).
- If the cell you want to execute an action on is in the fourth row and contains the value *Price* in the first column, you can address the row either with the index `4` (without quotes) or with the string *Price*.

Name	Data Type	Values	Default
Row Operator	String	equals not equals matches simple match	equals

Enter the operator you want to use for the selection of the row when you enter the row as a string.

- The *operator* parameter has four possible values.
- *"not equals"* looks for something that does *not exactly* match.
- *"equals"* looks for an *exact* match.

- Select "*simple match*" to use a simple match expression (→ page **??**) .
- Select "*matches*" to use a regular expression (→ page **??**) .
- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

Name	Data Type	Values	Default
Value	String	–	none

- Enter the value which is in the cell you want to drag.
- This parameter will be compared to cells in the row, using the operator provided.

Name	Data Type	Values	Default
Value Operator	String	equals not equals matches simple match	equals

Use this parameter to specify the operator used for the cell value.

- The *operator* parameter has four possible values.
- "*not equals*" looks for something that does *not exactly* match.
- "*equals*" looks for an *exact* match.
- Select "*simple match*" to use a simple match expression (→ page **??**) .
- Select "*matches*" to use a regular expression (→ page **??**) .
- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

Name	Data Type	Values	Default
Search Type	String	absolute relative	absolute

- The *search type* parameter has two options.
- "*absolute*" begins searching for the value at the first entry.
- Select "*relative*" to begin the search for the value after the currently selected entry. This allows you to search for the next occurence of a value, ignoring all elements before and including the selected entry.

2.2.7.13 Drop on Cell

(Table Component)

- This action drops a dragged item onto a cell you specify.
- The cell where the drop should occur is specified by its row and column values.
- The cursor is moved over the cell.
- The mouse button you specify is let loose, and the cell is dropped.
- Precede this action with a *drag* action

This action is unsupported for the HTML toolkit

Parameters

Name	Data Type	Values	Default
Row	String	–	none

Use this parameter to specify the row where the item should be dropped.

- In tables, you can address the row of a table either by using its index or by entering the value in the first column of the particular row.
- The first row is `1` (without quotes) and so on.
- The row of table headers at the top of the table can be addressed using the index `0` (without quotes).
- If the cell you want to execute an action on is in the fourth row and contains the value *Price* in the first column, you can address the row either with the index `4` (without quotes) or with the string *Price*.

Name	Data Type	Values	Default
Row Operator	String	equals not equals matches simple match	equals

Enter the operator you want to use for the selection of the row when you enter the row as a string.

- The *operator* parameter has four possible values.
- "*not equals*" looks for something that does *not exactly* match.
- "*equals*" looks for an *exact* match.

- Select "*simple match*" to use a simple match expression (→ page **??**) .
- Select "*matches*" to use a regular expression (→ page **??**) .
- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

Name	Data Type	Values	Default
Column	String	–	none

Use this parameter to specify the column where the item should be dropped.

- In tables, you can address the header of a table either by using its index or by entering its title.
- The first column is `1` (without quotes) and so on. The index `0` (without quotes) refers to the header itself.
- If the cell you want to execute an action on is in the fourth column and is called *Price*, you can address the column either with the index `4` (without quotes) or with the string *Price*.
- In JavaFX AUT's, you can also enter a path to the column to test tables that have nested columns (i.e. where a column has child columns). Enter the column as a path, separated with a forward slash, e.g. `Address/Street`. You can use indexpaths and textpaths for columns in JavaFX tables.

Name	Data Type	Values	Default
Column Operator	String	equals not equals matches simple match	equals

Enter the operator you want to use for the selection of the column when you enter the column as a string.

- The *operator* parameter has four possible values.
- "*not equals*" looks for something that does *not exactly* match.
- "*equals*" looks for an *exact* match.
- Select "*simple match*" to use a simple match expression (→ page **??**) .
- Select "*matches*" to use a regular expression (→ page **??**) .
- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

Name	Data Type	Values	Default
x-position	Integer	–	50

- With this parameter, you can specify where in the component you should click.

- Use this parameter to enter a value for the x-axis (the horizontal position).
- Using the x-units parameter, you can specify whether the value should be in pixels or in percent.
- For example, if you enter `50` and then `percent` (without quotes) , then the click will be performed in the middle of the x-axis.
- All positions are calculated from the upper left of the component, with the positive x-axis stretching toward the right side of the component.

We recommend not using `0` (pixels and percent) or `100`(percent) positions.

Name	Data Type	Values	Default
x-units	String	percent pixel	percent

- Use this parameter to specify whether you want your values for the x-coordinate to be in pixels or percent.
- If you choose percent for this parameter, and `50` for the *x-position* parameter, the click will always be in the middle of the x-axis.

Name	Data Type	Values	Default
y-position	Integer	–	50

- With this parameter, you can specify where in the component you should click.
- Use this parameter to enter a value for the y-axis (the vertical position).
- Using the y-units parameter, you can specify whether the value should be in pixels or in percent.
- For example, if you enter `50` and then `percent` (without quotes), then the click will be performed in the middle of the y-axis.
- All positions are calculated from the upper left of the component, with the positive y-axis stretching toward the bottom of the component.

We recommend not using `0` (pixels and percent) or `100`(percent) positions.

Name	Data Type	Values	Default
y-units	String	percent pixel	percent

- Use this parameter to specify whether you want your values for the y-coordinate to be in pixels or percent.
- If you choose percent for this parameter, and `50` for the *y-position* parameter, the click will always be in the middle of the y-axis.

Name	Data Type	Values	Default
Delay before drop (milliseconds)	Integer	–	100

- Use this parameter to wait before dropping the item you have dragged.
- This can be useful to give the user interface time to scroll etc.
- Give the time to delay in milliseconds.

2.2.7.14 Drop on Cell from Column

`(Table Component)`

- This action drops a dragged item onto a cell you specify.
- The cell where the drop should occur is specified by its column and the value in the cell.
- The cursor is moved over the cell.
- The mouse button you specify is let loose, and the cell is dropped.
- Precede this action with a *drag* action

This action is unsupported for the HTML toolkit

Parameters

Name	Data Type	Values	Default
Column	String	–	none

Use this parameter to specify the column where the drop should occur.

- In tables, you can address the header of a table either by using its index or by entering its title.
- The first column is `1` (without quotes) and so on. The index `0` (without quotes) refers to the header itself.

- If the cell you want to execute an action on is in the fourth column and is called *Price*, you can address the column either with the index `4` (without quotes) or with the string *Price*.
- In JavaFX AUT's, you can also enter a path to the column to test tables that have nested columns (i.e. where a column has child columns). Enter the column as a path, separated with a forward slash, e.g. `Address/Street`. You can use indexpaths and textpaths for columns in JavaFX tables.

Name	Data Type	Values	Default
Column Operator	String	equals not equals matches simple match	equals

Enter the operator you want to use for the selection of the column when you enter the column as a string.

- The *operator* parameter has four possible values.
- "*not equals*" looks for something that does *not exactly* match.
- "*equals*" looks for an *exact* match.
- Select "*simple match*" to use a simple match expression (→ page **??**) .
- Select "*matches*" to use a regular expression (→ page **??**) .
- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

Name	Data Type	Values	Default
Value	String	–	none

- Enter the value which is in the cell you want to drop onto.
- This parameter will be compared to cells in the column, using the operator provided.

Name	Data Type	Values	Default
Value Operator	String	equals not equals matches simple match	equals

Use this parameter to specify the operator used for the cell value.

- The *operator* parameter has four possible values.
- "*not equals*" looks for something that does *not exactly* match.
- "*equals*" looks for an *exact* match.
- Select "*simple match*" to use a simple match expression (→ page **??**) .

- Select "*matches*" to use a regular expression (→ page **??**) .
- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

Name	Data Type	Values	Default
Search Type	String	absolute relative	absolute

- The *search type* parameter has two options.
- "*absolute*" begins searching for the value at the first entry.
- Select "*relative*" to begin the search for the value after the currently selected entry. This allows you to search for the next occurence of a value, ignoring all elements before and including the selected entry.

Name	Data Type	Values	Default
Delay before drop (milliseconds)	Integer	–	100

- Use this parameter to wait before dropping the item you have dragged.
- This can be useful to give the user interface time to scroll etc.
- Give the time to delay in milliseconds.

2.2.7.15 Drop on Cell from Row

`(Table Component)`

- This action drops a dragged item onto a cell you specify.
- The cell where the drop should occur is specified by its row and the value in the cell.
- The cursor is moved over the cell.
- The mouse button you specify is let loose, and the cell is dropped.
- Precede this action with a *drag* action

This action is unsupported for the HTML toolkit

Parameters

Name	Data Type	Values	Default
Row	String	–	none

Use this parameter to specify the row where the drop should occur.

- In tables, you can address the row of a table either by using its index or by entering the value in the first column of the particular row.
- The first row is `1` (without quotes) and so on.
- The row of table headers at the top of the table can be addressed using the index `0` (without quotes).
- If the cell you want to execute an action on is in the fourth row and contains the value *Price* in the first column, you can address the row either with the index `4` (without quotes) or with the string *Price*.

Name	Data Type	Values	Default
Row Operator	String	equals not equals matches simple match	equals

Enter the operator you want to use for the selection of the row when you enter the row as a string.

- The *operator* parameter has four possible values.
- "*not equals*" looks for something that does *not exactly* match.
- "*equals*" looks for an *exact* match.
- Select "*simple match*" to use a simple match expression (→ page **??**) .
- Select "*matches*" to use a regular expression (→ page **??**) .
- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

Name	Data Type	Values	Default
Value	String	–	none

- Enter the value which is in the cell you want to drop onto.
- This parameter will be compared to cells in the row, using the operator provided.

Name	Data Type	Values	Default
Value Operator	String	equals not equals matches simple match	equals

Use this parameter to specify the operator used for the cell value.

- The *operator* parameter has four possible values.
- "*not equals*" looks for something that does *not exactly* match.
- "*equals*" looks for an *exact* match.
- Select "*simple match*" to use a simple match expression (→ page **??**) .
- Select "*matches*" to use a regular expression (→ page **??**) .
- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

Name	Data Type	Values	Default
Search Type	String	absolute relative	absolute

- The *search type* parameter has two options.
- "*absolute*" begins searching for the value at the first entry.
- Select "*relative*" to begin the search for the value after the currently selected entry. This allows you to search for the next occurence of a value, ignoring all elements before and including the selected entry.

Name	Data Type	Values	Default
Delay before drop (milliseconds)	Integer	–	100

- Use this parameter to wait before dropping the item you have dragged.
- This can be useful to give the user interface time to scroll etc.
- Give the time to delay in milliseconds.

2.2.7.16 Input Text (Specify Cell)

`(Table Component)`

- Use this action to enter a text into a cell.
- You specify the text to enter and the cell to enter it into.

  This is achieved by:

  - Clicking once on the cell to activate it
  - If the click does not activate the cell, the cell is double-clicked

– Entering the text in the cell.

This action is unsupported for the HTML toolkit

Parameters

Name	Data Type	Values	Default
Text	String	–	none

- Use this parameter to specify the text you want to enter into the cell.

Name	Data Type	Values	Default
Row	String	–	none

- In tables, you can address the row of a table either by using its index or by entering the value in the first column of the particular row.
- The first row is `1` (without quotes) and so on.
- The row of table headers at the top of the table can be addressed using the index `0` (without quotes).
- If the cell you want to execute an action on is in the fourth row and contains the value *Price* in the first column, you can address the row either with the index `4` (without quotes) or with the string *Price*.

Name	Data Type	Values	Default
Row Operator	String	equals not equals matches simple match	equals

Enter the operator you want to use for the selection of the row when you enter the row as a string.

- The *operator* parameter has four possible values.
- "*not equals*" looks for something that does *not exactly* match.
- "*equals*" looks for an *exact* match.
- Select "*simple match*" to use a simple match expression (→ page **??**) .
- Select "*matches*" to use a regular expression (→ page **??**) .
- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

Name	Data Type	Values	Default
Column	String	–	none

Use this parameter to specify the column which contains the cell.

- In tables, you can address the header of a table either by using its index or by entering its title.
- The first column is `1` (without quotes) and so on. The index `0` (without quotes) refers to the header itself.
- If the cell you want to execute an action on is in the fourth column and is called *Price*, you can address the column either with the index `4` (without quotes) or with the string *Price*.
- In JavaFX AUT's, you can also enter a path to the column to test tables that have nested columns (i.e. where a column has child columns). Enter the column as a path, separated with a forward slash, e.g. `Address/Street`. You can use indexpaths and textpaths for columns in JavaFX tables.

Name	Data Type	Values	Default
Column Operator	String	equals not equals matches simple match	equals

Enter the operator you want to use for the selection of the column when you enter the column as a string.

- The *operator* parameter has four possible values.
- "*not equals*" looks for something that does *not exactly* match.
- "*equals*" looks for an *exact* match.
- Select "*simple match*" to use a simple match expression (→ page **??**) .
- Select "*matches*" to use a regular expression (→ page **??**) .
- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

2.2.7.17 Move

`(Table Component)`

- Use this action to move from a currently selected cell to another cell.
- You specify in which direction you want to move, and how far (number of cells).
- You also specify how many clicks should be made on the new cell, if any.

- This action is useful for navigating in a table.
- This is achieved by:
  - searching for the cell to move to
  - placing the cursor on this cell at the position you specify
  - clicking the amount of times you specify
- The movement is carried out relative to the cell where the cursor is.
- This is usually the selected cell, unless you have already carried out a 'move" action.
- If the cursor is, for any reason, not currently over a cell, the movement is carried out relative to the selected cell.

This action is unsupported for the HTML toolkit

Parameters

Name	Data Type	Values	Default
Direction	String	up down right left	right

- Use this parameter to define the direction you want to move in.
- Combined with the cell count, you can move any number of cells in any direction.

Name	Data Type	Values	Default
Number of Cells	Integer	–	1

- Use this parameter to specify how many cells you want to move in the given direction.

Name	Data Type	Values	Default
Number of Clicks	Integer	–	1

- Enter the amount of clicks that should be performed on the component or item.
- If you enter `0`, the mouse pointer will just be moved over the specified component or item.
- Entering `1` (without quotes) will single-click on the component or item.

- Entering `2` (without quotes) will double-click on the component or item.

Name	Data Type	Values	Default
x-position	Integer	–	50

- With this parameter, you can specify where in the component you should click.
- Use this parameter to enter a value for the x-axis (the horizontal position).
- Using the x-units parameter, you can specify whether the value should be in pixels or in percent.
- For example, if you enter `50` and then `percent` (without quotes) , then the click will be performed in the middle of the x-axis.
- All positions are calculated from the upper left of the component, with the positive x-axis stretching toward the right side of the component.

We recommend not using `0` (pixels and percent) or `100`(percent) as positions.

Name	Data Type	Values	Default
x-units	String	percent pixel	percent

- Use this parameter to specify whether you want your values for the x-coordinate to be in pixels or percent.
- If you choose percent for this parameter, and `50` for the *x-position* parameter, the click will always be in the middle of the x-axis.

Name	Data Type	Values	Default
y-position	Integer	–	50

- With this parameter, you can specify where in the component you should click.
- Use this parameter to enter a value for the y-axis (the vertical position).
- Using the y-units parameter, you can specify whether the value should be in pixels or in percent.
- For example, if you enter `50` and then `percent` (without quotes), then the click will be performed in the middle of the y-axis.
- All positions are calculated from the upper left of the component, with the positive y-axis stretching toward the bottom of the component.

We recommend not using `0` (pixels and percent) or `100`(percent) as positions.

Name	Data Type	Values	Default
y-units	String	percent pixel	percent

- Use this parameter to specify whether you want your values for the y-coordinate to be in pixels or percent.
- If you choose percent for this parameter, and `50` for the *y-position* parameter, the click will always be in the middle of the y-axis.

Name	Data Type	Values	Default
Extend Selection	String	yes no	no

- The *Extend Selection* parameter has two possible values.
- "*yes*" indicates the chosen item will be added to the current selection.
- Select "*no*" to begin a new selection with the chosen item.

2.2.7.18 Replace Text (Specify Cell)

`(Table Component)`

- Use this action to replace any text already in the cell.
- The text you want to enter is given as a parameter.
- This is achieved by:
  - Selecting the text in the cell.
  - Entering the text you specified.
- This means that any previous text in the component is deleted.

This action is unsupported for the HTML toolkit

Parameters

Name	Data Type	Values	Default
Text	String	–	none

- Use this parameter to specify the text you want to enter into the cell.
- Any previous text in the cell will be deleted.

Name	Data Type	Values	Default
Row	String	–	none

Use this parameter to specify the row which contains the cell whose text you want to replace.

- In tables, you can address the row of a table either by using its index or by entering the value in the first column of the particular row.
- The first row is `1` (without quotes) and so on.
- The row of table headers at the top of the table can be addressed using the index `0` (without quotes).
- If the cell you want to execute an action on is in the fourth row and contains the value *Price* in the first column, you can address the row either with the index `4` (without quotes) or with the string *Price*.

Name	Data Type	Values	Default
Row Operator	String	equals not equals matches simple match	equals

Enter the operator you want to use for the selection of the row when you enter the row as a string.

- The *operator* parameter has four possible values.
- "*not equals*" looks for something that does *not exactly* match.
- "*equals*" looks for an *exact* match.
- Select "*simple match*" to use a simple match expression (→ page **??**) .
- Select "*matches*" to use a regular expression (→ page **??**) .
- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

Name	Data Type	Values	Default
Column	String	–	none

Use this parameter to specify the column which contains the cell.

- In tables, you can address the header of a table either by using its index or by entering its title.
- The first column is `1` (without quotes) and so on. The index `0` (without quotes) refers to the header itself.

- If the cell you want to execute an action on is in the fourth column and is called *Price*, you can address the column either with the index `4` (without quotes) or with the string *Price*.
- In JavaFX AUT's, you can also enter a path to the column to test tables that have nested columns (i.e. where a column has child columns). Enter the column as a path, separated with a forward slash, e.g. `Address/Street`. You can use indexpaths and textpaths for columns in JavaFX tables.

Name	Data Type	Values	Default
Column Operator	String	equals not equals matches simple match	equals

Enter the operator you want to use for the selection of the column when you enter the column as a string.

- The *operator* parameter has four possible values.
- "*not equals*" looks for something that does *not exactly* match.
- "*equals*" looks for an *exact* match.
- Select "*simple match*" to use a simple match expression (→ page **??**) .
- Select "*matches*" to use a regular expression (→ page **??**) .
- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

2.2.7.19 Select Cell

`(Table Component)`

- Use this action to select a cell.
- You specify the cell by giving the row number and column header or index.
- This is achieved by:
  - Searching for the cell.
  - Placing the cursor in indicated cell, at the position you give.
  - Clicking as many times as you specify.

This action is unsupported for the HTML toolkit

Parameters

Name	Data Type	Values	Default
Row	String	–	none

Use this parameter to specify the row which contains the cell you want to select.

- In tables, you can address the row of a table either by using its index or by entering the value in the first column of the particular row.
- The first row is `1` (without quotes) and so on.
- The row of table headers at the top of the table can be addressed using the index `0` (without quotes).
- If the cell you want to execute an action on is in the fourth row and contains the value *Price* in the first column, you can address the row either with the index `4` (without quotes) or with the string *Price*.

Name	Data Type	Values	Default
Row Operator	String	equals not equals matches simple match	equals

Enter the operator you want to use for the selection of the row when you enter the row as a string.

- The *operator* parameter has four possible values.
- "*not equals*" looks for something that does *not exactly* match.
- "*equals*" looks for an *exact* match.
- Select "*simple match*" to use a simple match expression (→ page **??**) .
- Select "*matches*" to use a regular expression (→ page **??**) .
- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

Name	Data Type	Values	Default
Column	String	–	none

Use this parameter to enter the column number or header title which contains the cell you want to select.

- In tables, you can address the header of a table either by using its index or by entering its title.

- The first column is `1` (without quotes) and so on. The index `0` (without quotes) refers to the header itself.
- If the cell you want to execute an action on is in the fourth column and is called *Price*, you can address the column either with the index `4` (without quotes) or with the string *Price*.
- In JavaFX AUT's, you can also enter a path to the column to test tables that have nested columns (i.e. where a column has child columns). Enter the column as a path, separated with a forward slash, e.g. `Address/Street`. You can use indexpaths and textpaths for columns in JavaFX tables.

Name	Data Type	Values	Default
Column Operator	String	equals not equals matches simple match	equals

Enter the operator you want to use for the selection of the column when you enter the column as a string.

- The *operator* parameter has four possible values.
- "*not equals*" looks for something that does *not exactly* match.
- "*equals*" looks for an *exact* match.
- Select "*simple match*" to use a simple match expression (→ page **??**) .
- Select "*matches*" to use a regular expression (→ page **??**) .
- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

Name	Data Type	Values	Default
Number of Clicks	Integer	–	1

- Enter the amount of clicks that should be performed on the component or item.
- If you enter `0`, the mouse pointer will just be moved over the specified component or item.
- Entering `1` (without quotes) will single-click on the component or item.
- Entering `2` (without quotes) will double-click on the component or item.

Name	Data Type	Values	Default
x-position	Integer	–	50

- With this parameter, you can specify where in the component you should click.
- Use this parameter to enter a value for the x-axis (the horizontal position).

- Using the x-units parameter, you can specify whether the value should be in pixels or in percent.
- For example, if you enter `50` and then `percent` (without quotes) , then the click will be performed in the middle of the x-axis.
- All positions are calculated from the upper left of the component, with the positive x-axis stretching toward the right side of the component.

We recommend not using `0` (pixels and percent) or `100`(percent) as positions.

Name	Data Type	Values	Default
x-units	String	percent pixel	percent

- Use this parameter to specify whether you want your values for the x-coordinate to be in pixels or percent.
- If you choose percent for this parameter, and `50` for the *x-position* parameter, the click will always be in the middle of the x-axis.

Name	Data Type	Values	Default
y-position	Integer	–	50

- With this parameter, you can specify where in the component you should click.
- Use this parameter to enter a value for the y-axis (the vertical position).
- Using the y-units parameter, you can specify whether the value should be in pixels or in percent.
- For example, if you enter `50` and then `percent` (without quotes), then the click will be performed in the middle of the y-axis.
- All positions are calculated from the upper left of the component, with the positive y-axis stretching toward the bottom of the component.

We recommend not using `0` (pixels and percent) or `100`(percent) as positions.

Name	Data Type	Values	Default
y-units	String	percent pixel	percent

- Use this parameter to specify whether you want your values for the y-coordinate to be in pixels or percent.

- If you choose percent for this parameter, and `50` for the *y-position* parameter, the click will always be in the middle of the y-axis.

Name	Data Type	Values	Default
Extend Selection	String	yes no	no

- The *Extend Selection* parameter has two possible values.
- "*yes*" indicates the chosen item will be added to the current selection.
- Select "*no*" to begin a new selection with the chosen item.

Name	Data Type	Values	Default
Mouse Button	Integer	1 2 3	1

- Use this parameter to select which mouse button you want to click with.
  - `1` = left mouse button
  - `2` = middle mouse button
  - `3` = right mouse button
- The mouse button you use to click with can also have an effect on what happens.
- The button to click will depend on your AUT and what the click should achieve.

Do not enter quotes around the mouse button numbers, en them in plain text.

2.2.7.20 Select Value from Column

`(Table Component)`

- This action lets you select a cell in a table.
- To select the cell, you enter the column the cell is in and the value contained in the cell you want to select.
- Only the first cell which matches the criteria will be selected.

- This is achieved by clicking on the specified cell.

This action is unsupported for the HTML toolkit

Parameters

Name	Data Type	Values	Default
Column	String	–	none

Enter the number or header title of the column which contains the cell you want to select.

- In tables, you can address the header of a table either by using its index or by entering its title.
- The first column is `1` (without quotes) and so on. The index `0` (without quotes) refers to the header itself.
- If the cell you want to execute an action on is in the fourth column and is called *Price*, you can address the column either with the index `4` (without quotes) or with the string *Price*.
- In JavaFX AUT's, you can also enter a path to the column to test tables that have nested columns (i.e. where a column has child columns). Enter the column as a path, separated with a forward slash, e.g. `Address/Street`. You can use indexpaths and textpaths for columns in JavaFX tables.

Name	Data Type	Values	Default
Column Operator	String	equals not equals matches simple match	equals

Enter the operator you want to use for the selection of the column when you enter the column as a string.

- The *operator* parameter has four possible values.
- "*not equals*" looks for something that does *not exactly* match.
- "*equals*" looks for an *exact* match.
- Select "*simple match*" to use a simple match expression (→ page **??**) .
- Select "*matches*" to use a regular expression (→ page **??**) .
- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

Name	Data Type	Values	Default
Value	String	–	none

- Enter the value which is in the cell you want to select.

- This parameter will be compared to cells in the row, using the Operator provided.

Name	Data Type	Values	Default
Value Operator	String	equals not equals matches simple match	equals

Use this parameter to specify the operator used for the cell value.

- The *operator* parameter has four possible values.
- "*not equals*" looks for something that does *not exactly* match.
- "*equals*" looks for an *exact* match.
- Select "*simple match*" to use a simple match expression (→ page **??**) .
- Select "*matches*" to use a regular expression (→ page **??**) .
- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

Name	Data Type	Values	Default
Number of Clicks	Integer	–	1

- Enter the amount of clicks that should be performed on the component or item.
- If you enter `0`, the mouse pointer will just be moved over the specified component or item.
- Entering `1` (without quotes) will single-click on the component or item.
- Entering `2` (without quotes) will double-click on the component or item.

Name	Data Type	Values	Default
Extend Selection	String	yes no	no

- The *Extend Selection* parameter has two possible values.
- "*yes*" indicates the chosen item will be added to the current selection.
- Select "*no*" to begin a new selection with the chosen item.

Name	Data Type	Values	Default
Search Type	String	absolute relative	absolute

- The *search type* parameter has two options.
- "*absolute*" begins searching for the value at the first entry.
- Select "*relative*" to begin the search for the value after the currently selected entry. This allows you to search for the next occurence of a value, ignoring all elements before and including the selected entry.

Name	Data Type	Values	Default
Mouse Button	Integer	1 2 3	1

- Use this parameter to select which mouse button you want to click with.
  - `1` = left mouse button
  - `2` = middle mouse button
  - `3` = right mouse button
- The mouse button you use to click with can also have an effect on what happens.
- The button to click will depend on your AUT and what the click should achieve.

Do not enter quotes around the mouse button numbers, enter them in plain text.

2.2.7.21 Select Value from Row

`(Table Component)`

- This action lets you select a cell in a table.
- To select the cell, you enter the row number the cell is in and the value contained in the cell you want to select.
- Only the first cell which matches the criteria will be selected.
- This is achieved by clicking on the specified cell.

This action is unsupported for the HTML toolkit

Parameters

Name	Data Type	Values	Default
Row	String	–	none

Enter the number of the row which contains the cell you want to select.

- In tables, you can address the row of a table either by using its index or by entering the value in the first column of the particular row.
- The first row is `1` (without quotes) and so on.
- The row of table headers at the top of the table can be addressed using the index `0` (without quotes).
- If the cell you want to execute an action on is in the fourth row and contains the value *Price* in the first column, you can address the row either with the index `4` (without quotes) or with the string *Price*.

Name	Data Type	Values	Default
Row Operator	String	equals not equals matches simple match	equals

Enter the operator you want to use for the selection of the row when you enter the row as a string.

- The *operator* parameter has four possible values.
- "*not equals*" looks for something that does *not exactly* match.
- "*equals*" looks for an *exact* match.
- Select "*simple match*" to use a simple match expression (→ page **??**) .
- Select "*matches*" to use a regular expression (→ page **??**) .
- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

Name	Data Type	Values	Default
Value	String	–	none

- Enter the value which is in the cell you want to select.
- This parameter will be compared to cells in the row, using the Operator provided.

Name	Data Type	Values	Default
Value Operator	String	equals not equals matches simple match	equals

Use this parameter to specify the operator used for the cell value.

- The *operator* parameter has four possible values.
- "*not equals*" looks for something that does *not exactly* match.
- "*equals*" looks for an *exact* match.
- Select "*simple match*" to use a simple match expression (→ page **??**) .
- Select "*matches*" to use a regular expression (→ page **??**) .
- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

Name	Data Type	Values	Default
Number of Clicks	Integer	–	1

- Enter the amount of clicks that should be performed on the component or item.
- If you enter `0`, the mouse pointer will just be moved over the specified component or item.
- Entering `1` (without quotes) will single-click on the component or item.
- Entering `2` (without quotes) will double-click on the component or item.

Name	Data Type	Values	Default
Extend Selection	String	yes no	no

- The *Extend Selection* parameter has two possible values.
- "*yes*" indicates the chosen item will be added to the current selection.
- Select "*no*" to begin a new selection with the chosen item.

Name	Data Type	Values	Default
Search Type	String	absolute relative	absolute

- The *search type* parameter has two options.
- "*absolute*" begins searching for the value at the first entry.
- Select "*relative*" to begin the search for the value after the currently selected entry. This allows you to search for the next occurence of a value, ignoring all elements before and including the selected entry.

Name	Data Type	Values	Default
Mouse Button	Integer	1 2 3	1

- Use this parameter to select which mouse button you want to click with.
  - `1` = left mouse button
  - `2` = middle mouse button
  - `3` = right mouse button
- The mouse button you use to click with can also have an effect on what happens.
- The button to click will depend on your AUT and what the click should achieve.

Do not enter quotes around the mouse button numbers, en them in plain text.

2.2.7.22 Store Value (Specify Cell)

`(Table Component)`

- Use this action to read the value out of a cell so that you can use it as data for other Test Steps.

Parameters

Name	Data Type	Values	Default
Variable Name	Variable	–	none

- Enter the name you want to give to this variable.
- Variable names may only contain letters, numbers and underscores.
- You can then enter this variable name as data for other Test Steps.
- When you enter the variable name as data, place a dollar sign before it.
- The data associated with this variable name remains the same until the Project is closed, or until you overwrite the name by using it for another value.

Name	Data Type	Values	Default
Row	String	–	none

- In tables, you can address the row of a table either by using its index or by entering the value in the first column of the particular row.
- The first row is `1` (without quotes) and so on.

- The row of table headers at the top of the table can be addressed using the index `0` (without quotes).
- If the cell you want to execute an action on is in the fourth row and contains the value *Price* in the first column, you can address the row either with the index `4` (without quotes) or with the string *Price*.

Name	Data Type	Values	Default
Row Operator	String	equals not equals matches simple match	equals

Enter the operator you want to use for the selection of the row when you enter the row as a string.

- The *operator* parameter has four possible values.
- "*not equals*" looks for something that does *not exactly* match.
- "*equals*" looks for an *exact* match.
- Select "*simple match*" to use a simple match expression (→ page **??**) .
- Select "*matches*" to use a regular expression (→ page **??**) .
- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

Name	Data Type	Values	Default
Column	String	–	none

Use this parameter to enter the column number or header title which contains the cell whose data you want to store.

- In tables, you can address the header of a table either by using its index or by entering its title.
- The first column is `1` (without quotes) and so on. The index `0` (without quotes) refers to the header itself.
- If the cell you want to execute an action on is in the fourth column and is called *Price*, you can address the column either with the index `4` (without quotes) or with the string *Price*.
- In JavaFX AUT's, you can also enter a path to the column to test tables that have nested columns (i.e. where a column has child columns). Enter the column as a path, separated with a forward slash, e.g. `Address/Street`. You can use indexpaths and textpaths for columns in JavaFX tables.

Name	Data Type	Values	Default
Column Operator	String	equals not equals matches simple match	equals

Enter the operator you want to use for the selection of the column when you enter the column as a string.

- The *operator* parameter has four possible values.
- "*not equals*" looks for something that does *not exactly* match.
- "*equals*" looks for an *exact* match.
- Select "*simple match*" to use a simple match expression (→ page **??**) .
- Select "*matches*" to use a regular expression (→ page **??**) .
- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

2.2.7.23 Store Value at Mouse Position

`(Table Component)`

- Use this action to read the value in the cell where the mouse is currently placed so that you can use the value in the cell as data for other Test Steps.
- This can be used in conjunction with the action "*select cell*" with a click count of 0.

This action is unsupported for the HTML toolkit

Parameters

Name	Data Type	Values	Default
Variable Name	Variable	–	none

- Enter the name you want to give to this variable.
- Variable names may only contain letters, numbers and underscores.
- You can then enter this variable name as data for other Test Steps.
- When you enter the variable name as data, place a dollar sign before it.
- The data associated with this variable name remains the same until the Project is closed, or until you overwrite the name by using it for another value.

2.2.7.24 Toggle Checkbox at Mouse Position

`(Table Component)`

- Use this action to select or deselect a checkbox that is in the first column of a table.
- To use this action, the current mouse position must be over the correct cell. You can achieve this using the select actions, either with 0 clicks or with 1 click.
- If the checkbox is currently selected, this action will deselect it and vice-versa.
- This action is currently only supported for SWT/RCP AUT's.

This action has no parameters.

2.2.7.25 Toggle Checkbox in Selected Row

`(Table Component)`

- Use this action to select or deselect a checkbox that is in the first column of a table.
- To use this action, the row must already be selected.
- If the checkbox is currently selected, this action will deselect it and vice-versa.
- This action is currently only supported for SWT/RCP AUT's.

This action has no parameters.

2.2.8 Text Component

Description:
A *label* is used for displaying a small, descriptive text and/or an image.

Figure 2.13: *Label*

Mapping labels
In the Object Mapping Mode, a label to be mapped looks like this:

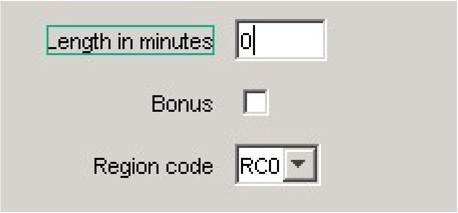


Figure 2.14: *Label*

Synopsis:

- Text Component (concrete)
  - Text Component (abstract)
    - Graphics Component (abstract)

New Actions

none

Inherited Actions

Name	Type	Parameters	Inherited from
Check Enablement (→ page 23)	c	Boolean:Enabled	Graphics Component (abstract)
Check Enablement of Context Menu Entry by Indexpath (→ page 24)	c	String:Indexpath Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)

Check Enablement of Context Menu Entry by Indexpath (Specify Position) (→ page 25)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Indexpath Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)
Check Enablement of Context Menu Entry by Textpath (→ page 28)	c	String:Textpath String:Operator Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)
Check Enablement of Context Menu Entry by Textpath (Specify Position) (→ page 29)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Textpath String:Operator Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)
Check Existence (→ page 32)	c	Boolean:Exists	Graphics Component (abstract)
Check Existence of Context Menu Entry by Indexpath (→ page 33)	c	String:Indexpath Boolean:Exists Integer:Mouse Button	Graphics Component (abstract)
Check Existence of Context Menu Entry by Indexpath (Specify Position) (→ page 34)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Indexpath Boolean:Exists Integer:Mouse Button	Graphics Component (abstract)

Check Existence of Context Menu Entry by Textpath (→ page 37)	c	String:Textpath String:Operator Boolean:Exists Integer:Mouse Button	Graphics Component (abstract)
Check Existence of Context Menu Entry by Textpath (Specify Position) (→ page 38)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Textpath String:Operator Boolean:Exists Integer:Mouse Button	Graphics Component (abstract)
Check Focus (→ page 41)	c	Boolean:Has Focus	Graphics Component (abstract)
Check Property (→ page 42)	c	String:Property Name String:Property Value String:Operator	Graphics Component (abstract)
Check Selection of Context Menu Entry by Indexpath (→ page 43)	c	String:Indexpath Boolean:Selected Integer:Mouse Button	Graphics Component (abstract)
Check Selection of Context Menu Entry by Indexpath (Specify Position) (→ page 45)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Indexpath Boolean:Selected Integer:Mouse Button	Graphics Component (abstract)

Check Selection of Context Menu Entry by Textpath (→ page 47)	c	String:Textpath String:Operator Boolean:Selected Integer:Mouse Button	Graphics Component (abstract)
Check Selection of Context Menu Entry by Textpath (Specify Position) (→ page 49)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Textpath String:Operator Boolean:Selected Integer:Mouse Button	Graphics Component (abstract)
Check Text (→ page 76)	c	String:Text String:Operator	Text Component (abstract)
Click (→ page 52)	e	Integer:Number of Clicks Integer:Mouse Button	Graphics Component (abstract)
Click in Component (→ page 53)	e	Integer:Number of Clicks Integer:Mouse Button Integer:x-position String:x-units Integer:y-position String:y-units	Graphics Component (abstract)
Drag (→ page 55)	e	Integer:Mouse Button String:Modifier Keys Integer:x-position String:x-units Integer:y-position String:y-units	Graphics Component (abstract)

Drop (→ page 58)	e	Integer:x-position String:x-units Integer:y-position String:y-units Integer:Delay before drop (milliseconds)	Graphics Component (abstract)
Select Context Menu Entry by Indexpath (→ page 60)	e	String:Indexpath Integer:Mouse Button	Graphics Component (abstract)
Select Context Menu Entry by Indexpath (Specify Position) (→ page 61)	e	Integer:x-position String:x-units Integer:y-position String:y-units String:Indexpath Integer:Mouse Button	Graphics Component (abstract)
Select Context Menu Entry by Textpath (→ page 64)	e	String:Textpath String:Operator Integer:Mouse Button	Graphics Component (abstract)
Select Context Menu Entry by Textpath (Specify Position) (→ page 66)	e	Integer:x-position String:x-units Integer:y-position String:y-units String:Menupath String:Operator Integer:Mouse Button	Graphics Component (abstract)

Show Text (→ page 69)	e	String:Text Integer:Text Size (in points) Integer:Time per Word (in milliseconds) Integer:Window Width (in pixels)	Graphics Component (abstract)
Store Property (→ page 70)	e	Variable:Variable Name String:Property Name	Graphics Component (abstract)
Store Value (→ page 77)	e	Variable:Variable Name	Text Component (abstract)
Wait for Component (→ page 71)	e	Integer:Timeout in ms Integer:Delay after Visibility	Graphics Component (abstract)

2.2.9 Text Input Component

Description:
This class of components describes generally all editable text areas.

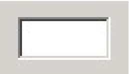


Figure 2.15: *Text Field*

Mapping text fields

In the Object Mapping Mode, a text field to be mapped looks like this:

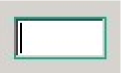


Figure 2.16: *Text Field*

Synopsis:

- Text Input Component (concrete)
  - Text Input Component (abstract)
    - ⋆ Text Component (abstract)
      - · Graphics Component (abstract)

New Actions

Name	Type	Parameters
Insert Text after Index (→ page 246)	e	String:Text Integer:Index
Insert Text before/after Pattern (→ page 247)	e	String:Text String:Pattern String:Operator Boolean:Insert after Pattern
Select All (→ page 248)	e	
Select Pattern (→ page 248)	e	String:Pattern String:Operator

Inherited Actions

Name	Type	Parameters	Inherited from
Check Editability (→ page 84)	c	Boolean:Editable	Text Input Component (abstract)

Check Enablement (→ page 23)	c	Boolean:Enabled	Graphics Component (abstract)
Check Enablement of Context Menu Entry by Indexpath (→ page 24)	c	String:Indexpath Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)
Check Enablement of Context Menu Entry by Indexpath (Specify Position) (→ page 25)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Indexpath Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)
Check Enablement of Context Menu Entry by Textpath (→ page 28)	c	String:Textpath String:Operator Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)
Check Enablement of Context Menu Entry by Textpath (Specify Position) (→ page 29)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Textpath String:Operator Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)
Check Existence (→ page 32)	c	Boolean:Exists	Graphics Component (abstract)
Check Existence of Context Menu Entry by Indexpath (→ page 33)	c	String:Indexpath Boolean:Exists Integer:Mouse Button	Graphics Component (abstract)

Check Existence of Context Menu Entry by Indexpath (Specify Position) (→ page 34)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Indexpath Boolean:Exists Integer:Mouse Button	Graphics Component (abstract)
Check Existence of Context Menu Entry by Textpath (→ page 37)	c	String:Textpath String:Operator Boolean:Exists Integer:Mouse Button	Graphics Component (abstract)
Check Existence of Context Menu Entry by Textpath (Specify Position) (→ page 38)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Textpath String:Operator Boolean:Exists Integer:Mouse Button	Graphics Component (abstract)
Check Focus (→ page 41)	c	Boolean:Has Focus	Graphics Component (abstract)
Check Property (→ page 42)	c	String:Property Name String:Property Value String:Operator	Graphics Component (abstract)
Check Selection of Context Menu Entry by Indexpath (→ page 43)	c	String:Indexpath Boolean:Selected Integer:Mouse Button	Graphics Component (abstract)
Check Selection of Context Menu Entry by Indexpath (Specify Position) (→ page 45)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Indexpath Boolean:Selected Integer:Mouse Button	Graphics Component (abstract)

Check Selection of Context Menu Entry by Textpath (→ page 47)	c	String:Textpath String:Operator Boolean:Selected Integer:Mouse Button	Graphics Component (abstract)
Check Selection of Context Menu Entry by Textpath (Specify Position) (→ page 49)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Textpath String:Operator Boolean:Selected Integer:Mouse Button	Graphics Component (abstract)
Check Text (→ page 76)	c	String:Text String:Operator	Text Component (abstract)
Click (→ page 52)	e	Integer:Number of Clicks Integer:Mouse Button	Graphics Component (abstract)
Click in Component (→ page 53)	e	Integer:Number of Clicks Integer:Mouse Button Integer:x-position String:x-units Integer:y-position String:y-units	Graphics Component (abstract)
Drag (→ page 55)	e	Integer:Mouse Button String:Modifier Keys Integer:x-position String:x-units Integer:y-position String:y-units	Graphics Component (abstract)

Drop (→ page 58)	e	Integer:x-position String:x-units Integer:y-position String:y-units Integer:Delay before drop (milliseconds)	Graphics Component (abstract)
Input Text (→ page 84)	e	String:Text	Text Input Component (abstract)
Replace Text (→ page 85)	e	String:Text	Text Input Component (abstract)
Select Context Menu Entry by Indexpath (→ page 60)	e	String:Indexpath Integer:Mouse Button	Graphics Component (abstract)
Select Context Menu Entry by Indexpath (Specify Position) (→ page 61)	e	Integer:x-position String:x-units Integer:y-position String:y-units String:Indexpath Integer:Mouse Button	Graphics Component (abstract)
Select Context Menu Entry by Textpath (→ page 64)	e	String:Textpath String:Operator Integer:Mouse Button	Graphics Component (abstract)
Select Context Menu Entry by Textpath (Specify Position) (→ page 66)	e	Integer:x-position String:x-units Integer:y-position String:y-units String:Menupath String:Operator Integer:Mouse Button	Graphics Component (abstract)

Show Text (→ page 69)	e	String:Text Integer:Text Size (in points) Integer:Time per Word (in milliseconds) Integer:Window Width (in pixels)	Graphics Component (abstract)
Store Property (→ page 70)	e	Variable:Variable Name String:Property Name	Graphics Component (abstract)
Store Value (→ page 77)	e	Variable:Variable Name	Text Component (abstract)
Wait for Component (→ page 71)	e	Integer:Timeout in ms Integer:Delay after Visibility	Graphics Component (abstract)

2.2.9.1 Insert Text after Index

`(Text Input Component)`

- Use this action to insert a text into the text area.
- You enter the text to be inserted, and the position at which it should be inserted.
- The position is given using the index.
- This is achieved by:
  - Positioning the cursor at the index you specify.
  - Entering the text.

Parameters

Name	Data Type	Values	Default
Text	String	–	none

- Enter the text to be inserted.

Name	Data Type	Values	Default
Index	Integer	–	none

- Enter the index after which the text will be inserted.

- The index is given as the number of characters up to the point where you want to insert your text.
- The first character is `1`.
- Spaces and symbols also count as characters.
- For example, if you want to enter a text after the string `My example`, enter the index `10`.

2.2.9.2 Insert Text before/after Pattern

`(Text Input Component)`

- Use this action to insert a text into the text area.
- You enter the text to be inserted and whether it should be inserted before or after the pattern you enter.
- This is achieved by:
  - Positioning the cursor at the position you specify.
  - Entering the text.
- If a pattern is not found, the text is not inserted.

Parameters

Name	Data Type	Values	Default
Text	String	–	none

- Enter the text you want to insert if the pattern is found.

Name	Data Type	Values	Default
Pattern	String	–	none

- Enter the pattern to search for. The search uses the Operator provided.
- If this pattern is found in the text, your text will be inserted before or after it, as defined.

Name	Data Type	Values	Default
Operator	String	equals not equals matches simple match	equals

- The *operator* parameter has four possible values.

- "*not equals*" looks for something that does *not exactly* match.
- "*equals*" looks for an *exact* match.
- Select "*simple match*" to use a simple match expression (→ page **??**) .
- Select "*matches*" to use a regular expression (→ page **??**) .
- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

Name	Data Type	Values	Default
Insert after Pattern	Boolean	true false	true

- Set to true to insert the text *after* the pattern.
- Set to false to insert the text *before* the pattern.

2.2.9.3 Select All

`(Text Input Component)`

- This action selects the entire text within the component.
- This is achieved by clicking three times within the text field.

This action is unsupported for the HTML toolkit

This action has no parameters.

2.2.9.4 Select Pattern

`(Text Input Component)`

- Use this action to select a given pattern within the text.
- If the pattern is not found, no text is selected.
- This is achieved without performing any clicks.

This action is unsupported for the HTML toolkit

Parameters

Name	Data Type	Values	Default
Pattern	String	–	none

- Enter the text you want to select if it is found.
- This parameter will be searched for within the component, using the Operator provided.

Name	Data Type	Values	Default
Operator	String	equals not equals matches simple match	equals

- The *operator* parameter has four possible values.
- "*not equals*" looks for something that does *not exactly* match.
- "*equals*" looks for an *exact* match.
- Select "*simple match*" to use a simple match expression (→ page **??**) .
- Select "*matches*" to use a regular expression (→ page **??**) .
- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

2.2.10 Tree Component

Description:

A Tree is a component with linked elements (nodes) and a hierarchical structure. One common example is the display of directory structures used in most file managers (e.g. *Windows Explorer*).

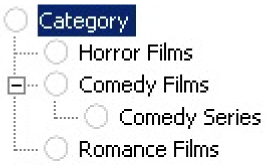


Figure 2.17: *Tree*

Because the forward slash (/) is a special symbol for trees, if you want to use a slash as part of your parameter value, you have to mask it. See the section later in this document (→ page **??**) for more details.

Mapping trees

In the Object Mapping Mode, a tree to be mapped looks like this:

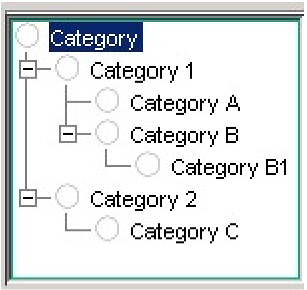


Figure 2.18: *Tree*

Actions on trees (as a hierarchical component) are not supported in the HTML toolkit. Individual nodes must be addressed as single links.

Virtual trees in SWT

The current support for trees has not been verified for virtual SWT trees.

Synopsis:

- Tree Component (concrete)

- Graphics Component (abstract)

New Actions

Name	Type	Parameters
Check Existence of Node by Indexpath (→ page 257)	c	String:Path Type Integer:Pre-ascend String:Indexpath Boolean:Exists
Check Existence of Node by Textpath (→ page 258)	c	String:Path Type Integer:Pre-ascend String:Textpath String:Operator Boolean:Exists
Check Text (Mouse Position) (→ page 260)	c	String:Text String:Operator
Check Text of Selected Node(s) (→ page 261)	c	String:Text String:Operator
Collapse Node by Indexpath (→ page 262)	e	String:Path Type Integer:Pre-ascend String:Indexpath
Collapse Node by Textpath (→ page 264)	e	String:Path Type Integer:Pre-ascend String:Textpath String:Operator
Drag Node by Indexpath (→ page 267)	e	Integer:Mouse Button String:Modifier Keys String:Path Type Integer:Pre-ascend String:Tree Indexpath
Drag Node by Textpath (→ page 269)	e	Integer:Mouse Button String:Modifier Keys String:Path Type Integer:Pre-ascend String:Tree Textpath String:Operator

Drop on Node by Indexpath (→ page 272)	e	String:Path Type Integer:Pre-ascend String:Tree Indexpath Integer:Delay before drop (milliseconds)
Drop on Node by Textpath (→ page 274)	e	String:Path Type Integer:Pre-ascend String:Tree Textpath String:Operator Integer:Delay before drop (milliseconds)
Expand Node by Indexpath (→ page 276)	e	String:Path Type Integer:Pre-ascend String:Indexpath
Expand Node by Textpath (→ page 277)	e	String:Path Type Integer:Pre-ascend String:Textpath String:Operator
Move (→ page 280)	e	String:Direction Integer:Number of Nodes Integer:Number of Clicks
Select Node by Indexpath (→ page 281)	e	String:Path Type Integer:Pre-ascend String:Indexpath Integer:Number of Clicks Integer:Mouse Button String:Extend Selection
Select Node by Textpath (→ page 283)	e	String:Path Type Integer:Pre-ascend String:Textpath String:Operator Integer:Number of Clicks Integer:Mouse Button String:Extend Selection
Store Text at Mouse Position (→ page 286)	e	Variable:Variable Name
Store Text of Selected Node (→ page 286)	e	Variable:Variable Name

Inherited Actions

Name	Type	Parameters	Inherited from
Check Enablement (→ page 23)	c	Boolean:Enabled	Graphics Component (abstract)
Check Enablement of Context Menu Entry by Indexpath (→ page 24)	c	String:Indexpath Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)
Check Enablement of Context Menu Entry by Indexpath (Specify Position) (→ page 25)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Indexpath Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)
Check Enablement of Context Menu Entry by Textpath (→ page 28)	c	String:Textpath String:Operator Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)
Check Enablement of Context Menu Entry by Textpath (Specify Position) (→ page 29)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Textpath String:Operator Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)

Check Existence (→ page 32)	c	Boolean:Exists	Graphics Component (abstract)
Check Existence of Context Menu Entry by Indexpath (→ page 33)	c	String:Indexpath Boolean:Exists Integer:Mouse Button	Graphics Component (abstract)
Check Existence of Context Menu Entry by Indexpath (Specify Position) (→ page 34)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Indexpath Boolean:Exists Integer:Mouse Button	Graphics Component (abstract)
Check Existence of Context Menu Entry by Textpath (→ page 37)	c	String:Textpath String:Operator Boolean:Exists Integer:Mouse Button	Graphics Component (abstract)
Check Existence of Context Menu Entry by Textpath (Specify Position) (→ page 38)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Textpath String:Operator Boolean:Exists Integer:Mouse Button	Graphics Component (abstract)
Check Focus (→ page 41)	c	Boolean:Has Focus	Graphics Component (abstract)
Check Property (→ page 42)	c	String:Property Name String:Property Value String:Operator	Graphics Component (abstract)

Check Selection of Context Menu Entry by Indexpath (→ page 43)	c	String:Indexpath Boolean:Selected Integer:Mouse Button	Graphics Component (abstract)
Check Selection of Context Menu Entry by Indexpath (Specify Position) (→ page 45)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Indexpath Boolean:Selected Integer:Mouse Button	Graphics Component (abstract)
Check Selection of Context Menu Entry by Textpath (→ page 47)	c	String:Textpath String:Operator Boolean:Selected Integer:Mouse Button	Graphics Component (abstract)
Check Selection of Context Menu Entry by Textpath (Specify Position) (→ page 49)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Textpath String:Operator Boolean:Selected Integer:Mouse Button	Graphics Component (abstract)
Click (→ page 52)	e	Integer:Number of Clicks Integer:Mouse Button	Graphics Component (abstract)
Click in Component (→ page 53)	e	Integer:Number of Clicks Integer:Mouse Button Integer:x-position String:x-units Integer:y-position String:y-units	Graphics Component (abstract)

Drag (→ page 55)	e	Integer:Mouse Button String:Modifier Keys Integer:x-position String:x-units Integer:y-position String:y-units	Graphics Component (abstract)
Drop (→ page 58)	e	Integer:x-position String:x-units Integer:y-position String:y-units Integer:Delay before drop (milliseconds)	Graphics Component (abstract)
Select Context Menu Entry by Indexpath (→ page 60)	e	String:Indexpath Integer:Mouse Button	Graphics Component (abstract)
Select Context Menu Entry by Indexpath (Specify Position) (→ page 61)	e	Integer:x-position String:x-units Integer:y-position String:y-units String:Indexpath Integer:Mouse Button	Graphics Component (abstract)
Select Context Menu Entry by Textpath (→ page 64)	e	String:Textpath String:Operator Integer:Mouse Button	Graphics Component (abstract)
Select Context Menu Entry by Textpath (Specify Position) (→ page 66)	e	Integer:x-position String:x-units Integer:y-position String:y-units String:Menupath String:Operator Integer:Mouse Button	Graphics Component (abstract)

Show Text (→ page 69)	e	String:Text Integer:Text Size (in points) Integer:Time per Word (in milliseconds) Integer:Window Width (in pixels)	Graphics Component (abstract)
Store Property (→ page 70)	e	Variable:Variable Name String:Property Name	Graphics Component (abstract)
Wait for Component (→ page 71)	e	Integer:Timeout in ms Integer:Delay after Visibility	Graphics Component (abstract)

2.2.10.1 Check Existence of Node by Indexpath

`(Tree Component)`

- This action checks that a node in the tree exists.
- You give the path to the node you want to check as an indexpath.
- For example, checking a textpath of "1/2/3" would check that the first node in the tree has at least 2 child nodes, and that the 2nd child node has at least 3 child nodes.

Parameters

Name	Data Type	Values	Default
Path Type	String	absolute relative	absolute

- The *path type* parameter has two values.
- "*absolute*" begins searching for the path at the root of the tree. Use this value when you enter the full path.
- Select "*relative*" to begin the search for the path at the currently selected node. This allows you to enter a partial path, ignoring all elements above the selected node.

Name	Data Type	Values	Default
Pre-ascend	Integer	–	0

- Use this parameter with a *relative path* to start the search above the currently selected node.
- The search will begin *n* nodes above the currently selected node, where *n* is the value of this parameter.
- For example, a value of 0 will simply begin with the currently selected node, and a value of 2 will begin with the parent of the parent of the currently selected node.
- A value can be given such that the search begins 1 level above the topmost visible level. This is then equivalent to an *absolute path*.
- If the search would begin any higher than this, the action fails: The tree node cannot be found.

The *Pre-ascend* parameter is simply ignored for *absolute paths*.

Name	Data Type	Values	Default
Indexpath	String	–	none

Enter the indexpath to the item you want to check. Make sure you give the whole path (either starting from the top of the tree, or at the position defined by the pre-ascend and path type parameters).

- Enter the path to the item as an indexpath.
- Use slash `'/'` as a path separator (to separate parent nodes from child nodes).
- For example, `1/2` (without quotes).

The first node is '1' (without quotes)

Name	Data Type	Values	Default
Exists	Boolean	true false	true

- Set to true if you expect the indexpath to exist.
- Set to false if you expect the indexpath *not* to exist.

2.2.10.2 Check Existence of Node by Textpath

`(Tree Component)`

- This action checks that a node in the tree exists.

- You give the path to the node you want to check as a textpath.
- For example, checking a textpath of "Dog/Bird/Cat" would check whether an entry "Cat" exists beneath an entry for "Bird" in the tree, which furthermore lies beneath an entry "Dog".

Parameters

Name	Data Type	Values	Default
Path Type	String	absolute relative	absolute

- The *path type* parameter has two values.
- "*absolute*" begins searching for the path at the root of the tree. Use this value when you enter the full path.
- Select "*relative*" to begin the search for the path at the currently selected node. This allows you to enter a partial path, ignoring all elements above the selected node.

Name	Data Type	Values	Default
Pre-ascend	Integer	–	0

- Use this parameter with a *relative path* to start the search above the currently selected node.
- The search will begin *n* nodes above the currently selected node, where *n* is the value of this parameter.
- For example, a value of 0 will simply begin with the currently selected node, and a value of 2 will begin with the parent of the parent of the currently selected node.
- A value can be given such that the search begins 1 level above the topmost visible level. This is then equivalent to an *absolute path*.
- If the search would begin any higher than this, the action fails: The tree node cannot be found.

The *Pre-ascend* parameter is simply ignored for *absolute paths*.

Name	Data Type	Values	Default
Textpath	String	–	none

Enter the textpath to the node you want to check. Make sure you give the whole path (either starting from the top of the tree, or at the position defined by the pre-ascend and path type parameters).

- Enter the path to the item as a textpath.

- Use slash `'/'` as a path separator (to separate parent nodes from child nodes).
- For example, `File/Open` or `Category/Horror` (without quotes).
- Either make sure that your path is written exactly as it appears in the interface, or use a regular expression to match the text.
- Each segment of the path will be used to find a corresponding node, using the operator provided.

Name	Data Type	Values	Default
Operator	String	equals not equals matches simple match	equals

- The *operator* parameter has four possible values.
- "*not equals*" looks for something that does *not exactly* match.
- "*equals*" looks for an *exact* match.
- Select "*simple match*" to use a simple match expression (→ page **??**) .
- Select "*matches*" to use a regular expression (→ page **??**) .
- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

When you use a regular expression for a textpath, bear in mind that each subpath is considered separately. The first path which corresponds to each subpath will be chosen. It is therefore advisable to make your regular expressions as unambiguous as possible.

Name	Data Type	Values	Default
Exists	Boolean	true false	true

- Set to true if you expect the textpath to exist.
- Set to false if you expect the textpath *not* to exist.

2.2.10.3 Check Text (Mouse Position)

`(Tree Component)`

- Use this action to check the text on a node in a tree based on the current position of the cursor.

- This can be used in conjunction with the action "*select node*" with a click count of 0.

Parameters

Name	Data Type	Values	Default
Text	String	–	none

- Enter the text you want to check.
- This parameter will be compared to the text on the node, using the operator provided.

Name	Data Type	Values	Default
Operator	String	equals not equals matches simple match	equals

- The *operator* parameter has four possible values.
- "*not equals*" looks for something that does *not exactly* match.
- "*equals*" looks for an *exact* match.
- Select "*simple match*" to use a simple match expression (→ page **??**) .
- Select "*matches*" to use a regular expression (→ page **??**) .
- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

2.2.10.4 Check Text of Selected Node(s)

`(Tree Component)`

- Use this action to check the text in a selected node or selected nodes in the tree.

Parameters

Name	Data Type	Values	Default
Text	String	–	none

- Use this parameter to enter the text (or regular expression (→ page **??**)) you want to check.
- If you want to check multiple values, separate the values with a comma `','`

- This parameter will be compared to the selected nodes, using the operator provided.

Name	Data Type	Values	Default
Operator	String	equals not equals matches simple match	equals

- The *operator* parameter has four possible values.
- "*not equals*" looks for something that does *not exactly* match.
- "*equals*" looks for an *exact* match.
- Select "*simple match*" to use a simple match expression (→ page **??**) .
- Select "*matches*" to use a regular expression (→ page **??**) .
- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

2.2.10.5 Collapse Node by Indexpath

`(Tree Component)`

- Use this action to collapse a tree or part of a tree along an indicated path.
- You enter the path as an indexpath.
- This is achieved by activating the command in the tree to collapse it.

Parameters

Name	Data Type	Values	Default
Path Type	String	absolute relative	absolute

- The *path type* parameter has two values.
- "*absolute*" begins searching for the path at the root of the tree. Use this value when you enter the full path.
- Select "*relative*" to begin the search for the path at the currently selected node. This allows you to enter a partial path, ignoring all elements above the selected node.

Name	Data Type	Values	Default
Pre-ascend	Integer	–	0

- Use this parameter with a *relative path* to start the search above the currently selected node.
- The search will begin *n* nodes above the currently selected node, where *n* is the value of this parameter.
- For example, a value of 0 will simply begin with the currently selected node, and a value of 2 will begin with the parent of the parent of the currently selected node.
- A value can be given such that the search begins 1 level above the topmost visible level. This is then equivalent to an *absolute path*.
- If the search would begin any higher than this, the action fails: The tree node cannot be found.

The *Pre-ascend* parameter is simply ignored for *absolute paths*.

Name	Data Type	Values	Default
Indexpath	String	–	none

Use this parameter to specify the indexpath of the subtree you want to collapse. Make sure you give the whole path (either starting from the top of the tree, or at the position defined by the pre-ascend and path type parameters).

- Enter the path to the item as an indexpath.
- Use slash `'/'` as a path separator (to separate parent nodes from child nodes).
- For example, `1/2` (without quotes).

The first node is '1' (without quotes)

Example:

- Your tree looks like this:

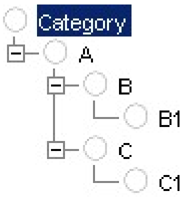


Figure 2.19: *Tree 1*

- You want to collapse node "C".

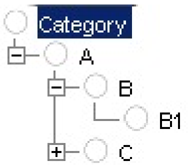


Figure 2.20: *Tree 2*

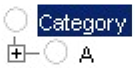


Figure 2.21: *Tree 3*

- Enter `1/1/2`:
- To collapse node *"A"*, enter `1/1`:
- To collapse a whole tree, enter `1`.

2.2.10.6 Collapse Node by Textpath

`(Tree Component)`

- Use this action to collapse a tree or part of a tree along an indicated path.
- You enter the path as a textpath.
- This is achieved by activating the command in the tree to collapse it.

Parameters

Name	Data Type	Values	Default
Path Type	String	absolute relative	absolute

- The *path type* parameter has two values.
- *"absolute"* begins searching for the path at the root of the tree. Use this value when you enter the full path.
- Select *"relative"* to begin the search for the path at the currently selected node. This allows you to enter a partial path, ignoring all elements above the selected node.

Name	Data Type	Values	Default
Pre-ascend	Integer	–	0

- Use this parameter with a *relative path* to start the search above the currently selected node.
- The search will begin *n* nodes above the currently selected node, where *n* is the value of this parameter.
- For example, a value of 0 will simply begin with the currently selected node, and a value of 2 will begin with the parent of the parent of the currently selected node.
- A value can be given such that the search begins 1 level above the topmost visible level. This is then equivalent to an *absolute path*.
- If the search would begin any higher than this, the action fails: The tree node cannot be found.

The *Pre-ascend* parameter is simply ignored for *absolute paths*.

Name	Data Type	Values	Default
Textpath	String	–	none

Use this parameter to specify the textpath of the subtree you want to collapse. Make sure you give the whole path (either starting from the top of the tree, or at the position defined by the pre-ascend and path type parameters).

- Enter the path to the item as a textpath.
- Use slash `'/'` as a path separator (to separate parent nodes from child nodes).
- For example, `File/Open` or `Category/Horror` (without quotes).
- Either make sure that your path is written exactly as it appears in the interface, or use a regular expression to match the text.
- Each segment of the path will be used to find a corresponding node, using the operator provided.

Example:

- Your tree looks like this:

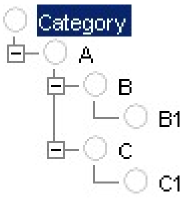


Figure 2.22: *Tree 1*

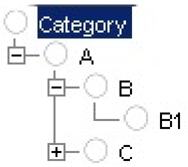


Figure 2.23: *Tree 2*

- You want to collapse node *"C"*.
- Enter `Category/A/C`:
- To collapse node *"A"*, enter `Category/A`:

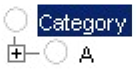


Figure 2.24: *Tree 3*

Name	Data Type	Values	Default
Operator	String	equals not equals matches simple match	equals

- The *operator* parameter has four possible values.
- *"not equals"* looks for something that does *not exactly* match.
- *"equals"* looks for an *exact* match.
- Select *"simple match"* to use a simple match expression (→ page **??**) .
- Select *"matches"* to use a regular expression (→ page **??**) .
- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

When you use a regular expression for a treepath, bear in mind that each subpath is considered separately. The first path in the tree which corresponds to each subpath will be chosen. It is therefore advisable to make your regular expressions as unambiguous as possible.

2.2.10.7 Drag Node by Indexpath

`(Tree Component)`

- This action drags the node you specify.
- You give the path to the node to be dragged using an indexpath.
- This is achieved by:
  - Moving the cursor over the component.
  - If the tree is not expanded, it is expanded.
  - Holding the mouse button you specify, and the item is dragged.
- You **must** follow this action with a *drop* action

Parameters

Name	Data Type	Values	Default
Mouse Button	Integer	1 2 3	1

- Use this parameter to select which mouse button you want to click with.
  - `1` = left mouse button
  - `2` = middle mouse button
  - `3` = right mouse button
- The mouse button you use to click with can also have an effect on what happens.
- The button to click will depend on your AUT and what the click should achieve.

Do not enter quotes around the mouse button numbers, en them in plain text.

Name	Data Type	Values	Default
Modifier Keys	String	none shift control alt meta cmd mod	none

- Use this parameter to specify a key or keys to be pressed at the same time as the action is being executed.
- You can use this parameter to specify the following keys, or combinations thereof:

  `shift`

  `control` This is the first modifier for Windows and Linux

  `cmd` This is the first modifier for Mac

  `mod` This is the platform-independent first modifier. It corresponds to »CONTROL« for Windows and Linux, and »CMD« for Mac.

  `meta`

  `alt`
- You can also select "*none*" if you do not need a modifier.
- You can enter multiple modifiers for a key combination by entering a space-separated list of modifiers, e.g. `alt shift`.

Do not use quotes around the modifiers, enter them in plain text

Name	Data Type	Values	Default
Path Type	String	absolute relative	absolute

- The *path type* parameter has two values.
- "*absolute*" begins searching for the path at the root of the tree. Use this value when you enter the full path.
- Select "*relative*" to begin the search for the path at the currently selected node. This allows you to enter a partial path, ignoring all elements above the selected node.

Name	Data Type	Values	Default
Pre-ascend	Integer	–	0

- Use this parameter with a *relative path* to start the search above the currently selected node.
- The search will begin *n* nodes above the currently selected node, where *n* is the value of this parameter.
- For example, a value of 0 will simply begin with the currently selected node, and a value of 2 will begin with the parent of the parent of the currently selected node.
- A value can be given such that the search begins 1 level above the topmost visible level. This is then equivalent to an *absolute path*.
- If the search would begin any higher than this, the action fails: The tree node cannot be found.

The *Pre-ascend* parameter is simply ignored for *absolute paths*.

Name	Data Type	Values	Default
Tree Indexpath	String	–	none

Make sure you give the whole path (either starting from the top of the tree, or at the position defined by the pre-ascend and path type parameters).

- Enter the path to the item as an indexpath.
- Use slash `'/'` as a path separator (to separate parent nodes from child nodes).
- For example, `1/2` (without quotes).

The first node is '1' (without quotes)

2.2.10.8 Drag Node by Textpath

`(Tree Component)`

- This action drags the node you specify.
- You give the path to the node to be dragged using a textpath.
- This is achieved by:
  - Moving the cursor over the component.
  - If the tree is not expanded, it is expanded.
  - Holding the mouse button you specify, and the item is dragged.
- You **must** follow this action with a *drop* action

Parameters

Name	Data Type	Values	Default
Mouse Button	Integer	1 2 3	1

- Use this parameter to select which mouse button you want to click with.

  - `1` = left mouse button
  - `2` = middle mouse button
  - `3` = right mouse button
- The mouse button you use to click with can also have an effect on what happens.
- The button to click will depend on your AUT and what the click should achieve.

Do not enter quotes around the mouse button numbers, enter them in plain text.

Name	Data Type	Values	Default
Modifier Keys	String	none shift control alt meta cmd mod	none

- Use this parameter to specify a key or keys to be pressed at the same time as the action is being executed.
- You can use this parameter to specify the following keys, or combinations thereof:

`shift`

`control` This is the first modifier for Windows and Linux

`cmd` This is the first modifier for Mac

`mod` This is the platform-independent first modifier. It corresponds to »CONTROL« for Windows and Linux, and »CMD« for Mac.

`meta`

`alt`

- You can also select "*none*" if you do not need a modifier.
- You can enter multiple modifiers for a key combination by entering a space-separated list of modifiers, e.g. `alt shift`.

Do not use quotes around the modifiers, enter them in plain text

Name	Data Type	Values	Default
Path Type	String	absolute relative	absolute

- The *path type* parameter has two values.

- "*absolute*" begins searching for the path at the root of the tree. Use this value when you enter the full path.
- Select "*relative*" to begin the search for the path at the currently selected node. This allows you to enter a partial path, ignoring all elements above the selected node.

Name	Data Type	Values	Default
Pre-ascend	Integer	–	0

- Use this parameter with a *relative path* to start the search above the currently selected node.
- The search will begin *n* nodes above the currently selected node, where *n* is the value of this parameter.
- For example, a value of 0 will simply begin with the currently selected node, and a value of 2 will begin with the parent of the parent of the currently selected node.
- A value can be given such that the search begins 1 level above the topmost visible level. This is then equivalent to an *absolute path*.
- If the search would begin any higher than this, the action fails: The tree node cannot be found.

The *Pre-ascend* parameter is simply ignored for *absolute paths*.

Name	Data Type	Values	Default
Tree Textpath	String	–	none

Use this parameter to specify the textpath. Make sure you give the whole path (either starting from the top of the tree, or at the position defined by the pre-ascend and path type parameters).

- Enter the path to the item as a textpath.
- Use slash `'/'` as a path separator (to separate parent nodes from child nodes).
- For example, `File/Open` or `Category/Horror` (without quotes).
- Either make sure that your path is written exactly as it appears in the interface, or use a regular expression to match the text.
- Each segment of the path will be used to find a corresponding node, using the operator provided.

Name	Data Type	Values	Default
Operator	String	equals not equals matches simple match	equals

- The *operator* parameter has four possible values.
- "*not equals*" looks for something that does *not exactly* match.
- "*equals*" looks for an *exact* match.
- Select "*simple match*" to use a simple match expression (→ page **??**) .
- Select "*matches*" to use a regular expression (→ page **??**) .
- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

2.2.10.9 Drop on Node by Indexpath

`(Tree Component)`

- This action drops a dragged item onto the node you specify.
- You give the path to the node you want to drop the item onto using an indexpath.
- This is achieved by:
  - Moving the cursor over the node.
  - If the tree is not expanded, it is expanded.
  - Letting loose the mouse button you specify.
- Precede this action with a *drag* action

Parameters

Name	Data Type	Values	Default
Path Type	String	absolute relative	absolute

- The *path type* parameter has two values.
- "*absolute*" begins searching for the path at the root of the tree. Use this value when you enter the full path.
- Select "*relative*" to begin the search for the path at the currently selected node. This allows you to enter a partial path, ignoring all elements above the selected node.

Name	Data Type	Values	Default
Pre-ascend	Integer	–	0

- Use this parameter with a *relative path* to start the search above the currently selected node.
- The search will begin *n* nodes above the currently selected node, where *n* is the value of this parameter.
- For example, a value of 0 will simply begin with the currently selected node, and a value of 2 will begin with the parent of the parent of the currently selected node.
- A value can be given such that the search begins 1 level above the topmost visible level. This is then equivalent to an *absolute path*.
- If the search would begin any higher than this, the action fails: The tree node cannot be found.

The *Pre-ascend* parameter is simply ignored for *absolute paths*.

Name	Data Type	Values	Default
Tree Indexpath	String	–	none

Make sure you give the whole path (either starting from the top of the tree, or at the position defined by the pre-ascend and path type parameters).

- Enter the path to the item as an indexpath.
- Use slash `'/'` as a path separator (to separate parent nodes from child nodes).
- For example, `1/2` (without quotes).

The first node is '1' (without quotes)

Name	Data Type	Values	Default
Delay before drop (milliseconds)	Integer	–	100

- Use this parameter to wait before dropping the item you have dragged.
- This can be useful to give the user interface time to scroll etc.
- Give the time to delay in milliseconds.

2.2.10.10 Drop on Node by Textpath

`(Tree Component)`

- This action drops a dragged item onto the node you specify.
- You give the path to the node you want to drop the item onto using a textpath.
- This is achieved by:
  - Moving the cursor over the node.
  - If the tree is not expanded, it is expanded.
  - Letting loose the mouse button you specify.
- Precede this action with a *drag* action

Parameters

Name	Data Type	Values	Default
Path Type	String	absolute relative	absolute

- The *path type* parameter has two values.
- "*absolute*" begins searching for the path at the root of the tree. Use this value when you enter the full path.
- Select "*relative*" to begin the search for the path at the currently selected node. This allows you to enter a partial path, ignoring all elements above the selected node.

Name	Data Type	Values	Default
Pre-ascend	Integer	–	0

- Use this parameter with a *relative path* to start the search above the currently selected node.
- The search will begin *n* nodes above the currently selected node, where *n* is the value of this parameter.
- For example, a value of 0 will simply begin with the currently selected node, and a value of 2 will begin with the parent of the parent of the currently selected node.
- A value can be given such that the search begins 1 level above the topmost visible level. This is then equivalent to an *absolute path*.
- If the search would begin any higher than this, the action fails: The tree node cannot be found.

The *Pre-ascend* parameter is simply ignored for *absolute paths*.

Name	Data Type	Values	Default
Tree Textpath	String	–	none

Use this parameter to specify the textpath. Make sure you give the whole path (either starting from the top of the tree, or at the position defined by the pre-ascend and path type parameters).

- Enter the path to the item as a textpath.
- Use slash `'/'` as a path separator (to separate parent nodes from child nodes).
- For example, `File/Open` or `Category/Horror` (without quotes).
- Either make sure that your path is written exactly as it appears in the interface, or use a regular expression to match the text.
- Each segment of the path will be used to find a corresponding node, using the operator provided.

Name	Data Type	Values	Default
Operator	String	equals not equals matches simple match	equals

- The *operator* parameter has four possible values.
- "*not equals*" looks for something that does *not exactly* match.
- "*equals*" looks for an *exact* match.
- Select "*simple match*" to use a simple match expression (→ page **??**) .
- Select "*matches*" to use a regular expression (→ page **??**) .
- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

Name	Data Type	Values	Default
Delay before drop (milliseconds)	Integer	–	100

- Use this parameter to wait before dropping the item you have dragged.
- This can be useful to give the user interface time to scroll etc.
- Give the time to delay in milliseconds.

2.2.10.11 Expand Node by Indexpath

`(Tree Component)`

- Use this action to expand a tree or part of a tree along an indicated path.
- You enter the path as an indexpath.
- This is achieved by activating the command in the tree to expand it.

Parameters

Name	Data Type	Values	Default
Path Type	String	absolute relative	absolute

- The *path type* parameter has two values.
- "*absolute*" begins searching for the path at the root of the tree. Use this value when you enter the full path.
- Select "*relative*" to begin the search for the path at the currently selected node. This allows you to enter a partial path, ignoring all elements above the selected node.

Name	Data Type	Values	Default
Pre-ascend	Integer	–	0

- Use this parameter with a *relative path* to start the search above the currently selected node.
- The search will begin *n* nodes above the currently selected node, where *n* is the value of this parameter.
- For example, a value of 0 will simply begin with the currently selected node, and a value of 2 will begin with the parent of the parent of the currently selected node.
- A value can be given such that the search begins 1 level above the topmost visible level. This is then equivalent to an *absolute path*.
- If the search would begin any higher than this, the action fails: The tree node cannot be found.

The *Pre-ascend* parameter is simply ignored for *absolute paths*.

Name	Data Type	Values	Default
Indexpath	String	–	none

Use this parameter to specify the indexpath of the subtree you want to expand. Make sure you give the whole path (either starting from the top of the tree, or at the position defined by the pre-ascend and path type parameters).

- Enter the path to the item as an indexpath.
- Use slash `'/'` as a path separator (to separate parent nodes from child nodes).
- For example, `1/2` (without quotes).

The first node is '1' (without quotes)

Example:

- Your tree looks like this:

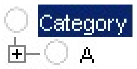


Figure 2.25: *Tree 3*

- You want to expand the tree to node A
- Enter `1/1`:

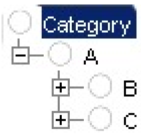


Figure 2.26: *Tree 4*

2.2.10.12 Expand Node by Textpath

`(Tree Component)`

- Use this action to expand a tree or part of a tree along an indicated path.
- You enter the path as a textpath.

- This is achieved by activating the command in the tree to expand it.

Parameters

Name	Data Type	Values	Default
Path Type	String	absolute relative	absolute

- The *path type* parameter has two values.
- "*absolute*" begins searching for the path at the root of the tree. Use this value when you enter the full path.
- Select "*relative*" to begin the search for the path at the currently selected node. This allows you to enter a partial path, ignoring all elements above the selected node.

Name	Data Type	Values	Default
Pre-ascend	Integer	–	0

- Use this parameter with a *relative path* to start the search above the currently selected node.
- The search will begin *n* nodes above the currently selected node, where *n* is the value of this parameter.
- For example, a value of 0 will simply begin with the currently selected node, and a value of 2 will begin with the parent of the parent of the currently selected node.
- A value can be given such that the search begins 1 level above the topmost visible level. This is then equivalent to an *absolute path*.
- If the search would begin any higher than this, the action fails: The tree node cannot be found.

The *Pre-ascend* parameter is simply ignored for *absolute paths*.

Name	Data Type	Values	Default
Textpath	String	–	none

Use this parameter to specify the textpath of the subtree you want to expand. Make sure you give the whole path (either starting from the top of the tree, or at the position defined by the pre-ascend and path type parameters).

- Enter the path to the item as a textpath.
- Use slash `'/'` as a path separator (to separate parent nodes from child nodes).
- For example, `File/Open` or `Category/Horror` (without quotes).

- Either make sure that your path is written exactly as it appears in the interface, or use a regular expression to match the text.
- Each segment of the path will be used to find a corresponding node, using the operator provided.

Example:

- Your tree looks like this:

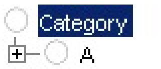


Figure 2.27: *Tree 3*

- You want to expand the tree to node A
- Enter `Category/A`:

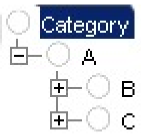


Figure 2.28: *Tree 4*

Name	Data Type	Values	Default
Operator	String	equals not equals matches simple match	equals

- The *operator* parameter has four possible values.
- "*not equals*" looks for something that does *not exactly* match.
- "*equals*" looks for an *exact* match.
- Select "*simple match*" to use a simple match expression (→ page **??**) .
- Select "*matches*" to use a regular expression (→ page **??**) .
- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

When you use a regular expression for a treepath, bear in mind that each subpath is considered separately. The first path in the tree which corresponds to each subpath will be chosen. It is therefore advisable to make your regular expressions as unambiguous as possible.

2.2.10.13 Move

`(Tree Component)`

- Use this action to move from a currently selected node to another node.
- You specify in which direction you want to move, and how far (number of nodes).
- You also specify how many clicks should be made on the new node, if any.
- This action is useful for navigating in a tree.
- This is achieved by:
  - searching for the node to move to
  - placing the cursor on this node
  - clicking the amount of times you specify
- The movement is carried out relative to the currently selected node.
- If no node is currently selected, the test will fail.

Parameters

Name	Data Type	Values	Default
Direction	String	up down next previous	up

Name	Data Type	Values	Default
Number of Nodes	Integer	–	1

- Use this parameter to specify how many nodes you want to move in the given direction.

Name	Data Type	Values	Default
Number of Clicks	Integer	–	1

- Enter the amount of clicks that should be performed on the component or item.
- If you enter `0`, the mouse pointer will just be moved over the specified component or item.
- Entering `1` (without quotes) will single-click on the component or item.
- Entering `2` (without quotes) will double-click on the component or item.

2.2.10.14 Select Node by Indexpath

`(Tree Component)`

- Use this item to select a node from a tree.
- You give the path to the node using an indexpath.
- This is achieved by:
  - Searching for the node you specify.
  - If the tree is not expanded, it is expanded.
  - Clicking on the node, the amount of times you specify.

Parameters

Name	Data Type	Values	Default
Path Type	String	absolute relative	absolute

- The *path type* parameter has two values.
- "*absolute*" begins searching for the path at the root of the tree. Use this value when you enter the full path.
- Select "*relative*" to begin the search for the path at the currently selected node. This allows you to enter a partial path, ignoring all elements above the selected node.

Name	Data Type	Values	Default
Pre-ascend	Integer	–	0

- Use this parameter with a *relative path* to start the search above the currently selected node.
- The search will begin *n* nodes above the currently selected node, where *n* is the value of this parameter.
- For example, a value of 0 will simply begin with the currently selected node, and a value of 2 will begin with the parent of the parent of the currently selected node.
- A value can be given such that the search begins 1 level above the topmost visible level. This is then equivalent to an *absolute path*.
- If the search would begin any higher than this, the action fails: The tree node cannot be found.

The *Pre-ascend* parameter is simply ignored for *absolute paths*.

Name	Data Type	Values	Default
Indexpath	String	–	none

Use this parameter to specify the node you want to select. Make sure you give the whole path (either starting from the top of the tree, or at the position defined by the pre-ascend and path type parameters).

- Enter the path to the item as an indexpath.
- Use slash `'/'` as a path separator (to separate parent nodes from child nodes).
- For example, `1/2` (without quotes).

The first node is '1' (without quotes)

Name	Data Type	Values	Default
Number of Clicks	Integer	–	1

- Enter the amount of clicks that should be performed on the component or item.
- If you enter `0`, the mouse pointer will just be moved over the specified component or item.
- Entering `1` (without quotes) will single-click on the component or item.
- Entering `2` (without quotes) will double-click on the component or item.

Name	Data Type	Values	Default
Mouse Button	Integer	1 2 3	1

- Use this parameter to select which mouse button you want to click with.
  - `1` = left mouse button
  - `2` = middle mouse button
  - `3` = right mouse button
- The mouse button you use to click with can also have an effect on what happens.
- The button to click will depend on your AUT and what the click should achieve.

Do not enter quotes around the mouse button numbers, enter them in plain text.

Name	Data Type	Values	Default
Extend Selection	String	yes no	no

- The *Extend Selection* parameter has two possible values.
- "*yes*" indicates the chosen item will be added to the current selection.
- Select "*no*" to begin a new selection with the chosen item.

2.2.10.15 Select Node by Textpath

`(Tree Component)`

- Use this item to select a node from a tree.
- You give the path to the node using a textpath.
- This is achieved by:
  - Searching for the node you specify.
  - If the tree is not expanded, it is expanded.
  - Clicking on the node, the amount of times you specify.

Parameters

Name	Data Type	Values	Default
Path Type	String	absolute relative	absolute

- The *path type* parameter has two values.
- "*absolute*" begins searching for the path at the root of the tree. Use this value when you enter the full path.
- Select "*relative*" to begin the search for the path at the currently selected node. This allows you to enter a partial path, ignoring all elements above the selected node.

Name	Data Type	Values	Default
Pre-ascend	Integer	–	0

- Use this parameter with a *relative path* to start the search above the currently selected node.
- The search will begin *n* nodes above the currently selected node, where *n* is the value of this parameter.

- For example, a value of 0 will simply begin with the currently selected node, and a value of 2 will begin with the parent of the parent of the currently selected node.
- A value can be given such that the search begins 1 level above the topmost visible level. This is then equivalent to an *absolute path*.
- If the search would begin any higher than this, the action fails: The tree node cannot be found.

The *Pre-ascend* parameter is simply ignored for *absolute paths*.

Name	Data Type	Values	Default
Textpath	String	–	none

Use this parameter to specify the node you want to select. Make sure you give the whole path (either starting from the top of the tree, or at the position defined by the pre-ascend and path type parameters).

- Enter the path to the item as a textpath.
- Use slash `'/'` as a path separator (to separate parent nodes from child nodes).
- For example, `File/Open` or `Category/Horror` (without quotes).
- Either make sure that your path is written exactly as it appears in the interface, or use a regular expression to match the text.
- Each segment of the path will be used to find a corresponding node, using the operator provided.

Name	Data Type	Values	Default
Operator	String	equals not equals matches simple match	equals

- The *operator* parameter has four possible values.
- "*not equals*" looks for something that does *not exactly* match.
- "*equals*" looks for an *exact* match.
- Select "*simple match*" to use a simple match expression (→ page **??**) .
- Select "*matches*" to use a regular expression (→ page **??**) .
- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

When you use a regular expression for a treepath, bear in mind that each subpath is considered separately. The first path in the tree which corresponds to each subpath will be chosen. It is therefore advisable to make your regular expressions as unambiguous as possible.

Name	Data Type	Values	Default
Number of Clicks	Integer	–	1

- Enter the amount of clicks that should be performed on the component or item.
- If you enter `0`, the mouse pointer will just be moved over the specified component or item.
- Entering `1` (without quotes) will single-click on the component or item.
- Entering `2` (without quotes) will double-click on the component or item.

Name	Data Type	Values	Default
Mouse Button	Integer	1 2 3	1

- Use this parameter to select which mouse button you want to click with.
  - `1` = left mouse button
  - `2` = middle mouse button
  - `3` = right mouse button
- The mouse button you use to click with can also have an effect on what happens.
- The button to click will depend on your AUT and what the click should achieve.

Do not enter quotes around the mouse button numbers, en them in plain text.

Name	Data Type	Values	Default
Extend Selection	String	yes no	no

- The *Extend Selection* parameter has two possible values.
- "*yes*" indicates the chosen item will be added to the current selection.
- Select "*no*" to begin a new selection with the chosen item.

2.2.10.16 Store Text at Mouse Position

`(Tree Component)`

- Use this action to read the value on the node where the mouse is currently placed so that you can use the value of the node as data for other Test Steps.
- This can be used in conjunction with the action "*select node*" with a click count of 0.

Parameters

Name	Data Type	Values	Default
Variable Name	Variable	–	none

- Enter the name you want to give to this variable.
- Variable names may only contain letters, numbers and underscores.
- You can then enter this variable name as data for other Test Steps.
- When you enter the variable name as data, place a dollar sign before it.
- The data associated with this variable name remains the same until the Project is closed, or until you overwrite the name by using it for another value.

2.2.10.17 Store Text of Selected Node

`(Tree Component)`

- Use this action to read the value from the currently selected node so that you can use the value as data for other Test Steps.

Parameters

Name	Data Type	Values	Default
Variable Name	Variable	–	none

- Enter the name you want to give to this variable.
- Variable names may only contain letters, numbers and underscores.
- You can then enter this variable name as data for other Test Steps.

- When you enter the variable name as data, place a dollar sign before it.
- The data associated with this variable name remains the same until the Project is closed, or until you overwrite the name by using it for another value.

Used By

(SWT) Tree (swt)
(SWT) Tree Table (swt)

2.3 gef Toolkit

The GEF toolkit can be used to test components within a figure canvas in RCP applications. You must set the toolkit to *RCP* in the Project properties to be able to use the GEF actions.

2.3.1 Figure Canvas

Description:

- The figure canvas (Figure 2.29 → page 288) is an editor where figures, connections and anchors are displayed.

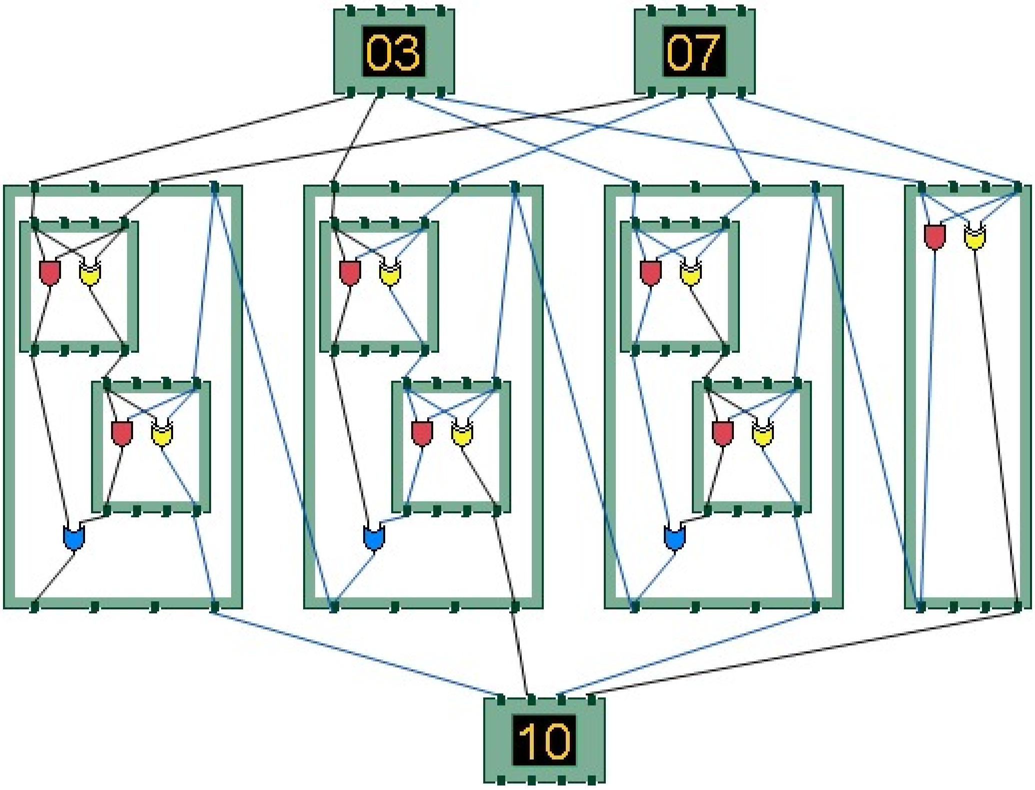


Figure 2.29: *Figure Canvas*

- Testing GEF components involves mapping this figure canvas and locating the figures based on their textpath.
- Use the GEF Inspector View to find out the textpath of the figures in the canvas (→*User Manual* p. 277).
- The figure canvas itself contains various items which can be addressed.
- Individual figures (Figure 2.30 → page 289).
- Tools (Figure 2.31 → page 290).
- Connections (Figure 2.32 → page 290).

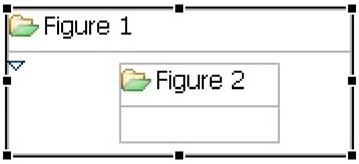
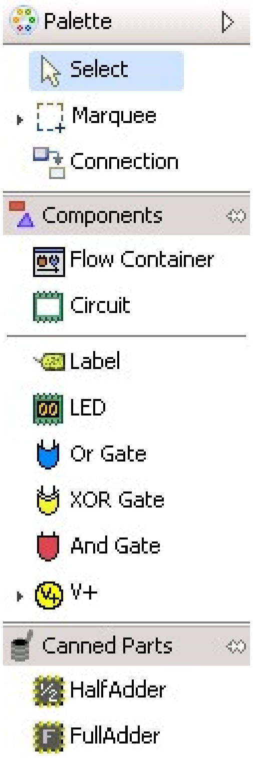


Figure 2.30: *A selected figure containing another figure*

- Connection anchors. These are the points where connections join to figures. Connection anchors are considered as figures. As such, they can be addressed (clicked, checked etc) using a textpath.



Palette
Select
Marquee
Connection
Components
Flow Container
Circuit
Label
LED
Or Gate
XOR Gate
And Gate
V+
Canned Parts
HalfAdder
FullAdder

Figure 2.31: *Various tools on the palette*

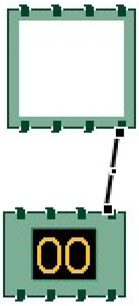


Figure 2.32: *A connection between two figures*

Synopsis:

- Figure Canvas (gef)
  - Graphics Component (abstract)

New Actions

Name	Type	Parameters
Check Figure Existence (→ page 296)	e	String:Textpath String:Operator Boolean:Exists
Check Figure Property (→ page 297)	c	String:Textpath String:Text Operator String:Property Name String:Property Value String:Value Operator
Check Tool Existence (→ page 299)	e	String:Textpath String:Operator Boolean:Exists
Click Connection (→ page 301)	e	String:Source Text Path String:Source Operator String:Target Text Path String:Target Operator Integer:Number of Clicks Integer:Mouse Button
Click Figure (→ page 303)	e	String:Textpath String:Operator Integer:Number of Clicks Integer:Mouse Button
Click in Figure (→ page 304)	e	String:Textpath String:Operator Integer:Number of Clicks Integer:Mouse Button Integer:x-position String:x-units Integer:y-position String:y-units

Drag Figure (→ page 307)	e	String:Textpath String:Operator Integer:Mouse Button String:Modifier Keys Integer:x-position String:x-units Integer:y-position String:y-units
Drop on Figure (→ page 311)	e	String:Textpath String:Operator Integer:x-position String:x-units Integer:y-position String:y-units Integer:Delay before drop (milliseconds)
Select Tool (→ page 313)	e	String:Textpath String:Operator Integer:Number of Clicks

Inherited Actions

Name	Type	Parameters	Inherited from
Check Enablement (→ page 23)	c	Boolean:Enabled	Graphics Component (abstract)
Check Enablement of Context Menu Entry by Indexpath (→ page 24)	c	String:Indexpath Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)
Check Enablement of Context Menu Entry by Indexpath (Specify Position) (→ page 25)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Indexpath Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)

Check Enablement of Context Menu Entry by Textpath (→ page 28)	c	String:Textpath String:Operator Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)
Check Enablement of Context Menu Entry by Textpath (Specify Position) (→ page 29)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Textpath String:Operator Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)
Check Existence (→ page 32)	c	Boolean:Exists	Graphics Component (abstract)
Check Existence of Context Menu Entry by Indexpath (→ page 33)	c	String:Indexpath Boolean:Exists Integer:Mouse Button	Graphics Component (abstract)
Check Existence of Context Menu Entry by Indexpath (Specify Position) (→ page 34)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Indexpath Boolean:Exists Integer:Mouse Button	Graphics Component (abstract)
Check Existence of Context Menu Entry by Textpath (→ page 37)	c	String:Textpath String:Operator Boolean:Exists Integer:Mouse Button	Graphics Component (abstract)
Check Existence of Context Menu Entry by Textpath (Specify Position) (→ page 38)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Textpath String:Operator Boolean:Exists Integer:Mouse Button	Graphics Component (abstract)

Check Focus (→ page 41)	c	Boolean:Has Focus	Graphics Component (abstract)
Check Property (→ page 42)	c	String:Property Name String:Property Value String:Operator	Graphics Component (abstract)
Check Selection of Context Menu Entry by Indexpath (→ page 43)	c	String:Indexpath Boolean:Selected Integer:Mouse Button	Graphics Component (abstract)
Check Selection of Context Menu Entry by Indexpath (Specify Position) (→ page 45)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Indexpath Boolean:Selected Integer:Mouse Button	Graphics Component (abstract)
Check Selection of Context Menu Entry by Textpath (→ page 47)	c	String:Textpath String:Operator Boolean:Selected Integer:Mouse Button	Graphics Component (abstract)
Check Selection of Context Menu Entry by Textpath (Specify Position) (→ page 49)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Textpath String:Operator Boolean:Selected Integer:Mouse Button	Graphics Component (abstract)

Click (→ page 52)	e	Integer:Number of Clicks Integer:Mouse Button	Graphics Component (abstract)
Click in Component (→ page 53)	e	Integer:Number of Clicks Integer:Mouse Button Integer:x-position String:x-units Integer:y-position String:y-units	Graphics Component (abstract)
Drag (→ page 55)	e	Integer:Mouse Button String:Modifier Keys Integer:x-position String:x-units Integer:y-position String:y-units	Graphics Component (abstract)
Drop (→ page 58)	e	Integer:x-position String:x-units Integer:y-position String:y-units Integer:Delay before drop (milliseconds)	Graphics Component (abstract)
Select Context Menu Entry by Indexpath (→ page 60)	e	String:Indexpath Integer:Mouse Button	Graphics Component (abstract)
Select Context Menu Entry by Indexpath (Specify Position) (→ page 61)	e	Integer:x-position String:x-units Integer:y-position String:y-units String:Indexpath Integer:Mouse Button	Graphics Component (abstract)

Select Context Menu Entry by Textpath (→ page 64)	e	String:Textpath String:Operator Integer:Mouse Button	Graphics Component (abstract)
Select Context Menu Entry by Textpath (Specify Position) (→ page 66)	e	Integer:x-position String:x-units Integer:y-position String:y-units String:Menupath String:Operator Integer:Mouse Button	Graphics Component (abstract)
Show Text (→ page 69)	e	String:Text Integer:Text Size (in points) Integer:Time per Word (in milliseconds) Integer:Window Width (in pixels)	Graphics Component (abstract)
Store Property (→ page 70)	e	Variable:Variable Name String:Property Name	Graphics Component (abstract)
Wait for Component (→ page 71)	e	Integer:Timeout in ms Integer:Delay after Visibility	Graphics Component (abstract)

2.3.1.1 Check Figure Existence

`(Figure Canvas)`

- Use this action to check whether a figure exists on the canvas.
- You identify the figure based on its textpath.

Parameters

Name	Data Type	Values	Default
Textpath	String	–	none

Use this parameter to enter the textpath to the figure you want to check.

- Enter the path to the item as a textpath.
- Use slash `'/'` as a path separator (to separate parent nodes from child nodes).
- For example, `File/Open` or `Category/Horror` (without quotes).
- Either make sure that your path is written exactly as it appears in the interface, or use a regular expression to match the text.
- Each segment of the path will be used to find a corresponding node, using the operator provided.

Name	Data Type	Values	Default
Operator	String	equals not equals matches simple match	equals

- The *operator* parameter has four possible values.
- "*not equals*" looks for something that does *not exactly* match.
- "*equals*" looks for an *exact* match.
- Select "*simple match*" to use a simple match expression (→ page **??**) .
- Select "*matches*" to use a regular expression (→ page **??**) .
- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

Name	Data Type	Values	Default
Exists	Boolean	true false	true

- Enter *true* if you expect the figure to exist.
- Enter *false* if you expect the figure *not* to exist.

2.3.1.2 Check Figure Property

`(Figure Canvas)`

- Use this action to check a specific property of a figure.

If you want the check a property of the figure canvas as a whole, use the action *Check Property* on the *Graphics Component* component.

- You can check whether a value you enter for a given property matches the actual value for the property.
- Enter the name of the property you want to check, and then what value it should have.

If the property is *private*, it cannot be checked, as the remote control component has no access to it.

Parameters

Name	Data Type	Values	Default
Textpath	String	–	none

- Enter the path to the item as a textpath.
- Use slash `'/'` as a path separator (to separate parent nodes from child nodes).
- For example, `File/Open` or `Category/Horror` (without quotes).
- Either make sure that your path is written exactly as it appears in the interface, or use a regular expression to match the text.
- Each segment of the path will be used to find a corresponding node, using the operator provided.

Use the GEF Inspector View to find out the textpath of items on the canvas (→*User Manual* p. 277).

Name	Data Type	Values	Default
Text Operator	String	equals not equals matches simple match	equals

Use this parameter to enter the operator you wish to use for the textpath to the figure.

- The *operator* parameter has four possible values.
- "*not equals*" looks for something that does *not exactly* match.
- "*equals*" looks for an *exact* match.
- Select "*simple match*" to use a simple match expression (→ page **??**) .
- Select "*matches*" to use a regular expression (→ page **??**) .
- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

When you use a regular expression for a textpath, bear in mind that each subpath is considered separately. The first path which corresponds to each subpath will be chosen. It is therefore advisable to make your regular expressions as unambiguous as possible.

Name	Data Type	Values	Default
Property Name	String	–	none

- Enter the name of the property you want to check.
- For example, to check the width, enter `width`.

Name	Data Type	Values	Default
Property Value	String	–	none

- Enter the value you expect the property to have.
- For `width` and `height` properties, enter the value in pixels.

Name	Data Type	Values	Default
Value Operator	String	equals not equals matches simple match	equals

Use this paramter to specify the operator you wish to use for the Value parameter.

- The *operator* parameter has four possible values.
- "*not equals*" looks for something that does *not exactly* match.
- "*equals*" looks for an *exact* match.
- Select "*simple match*" to use a simple match expression (→ page **??**) .
- Select "*matches*" to use a regular expression (→ page **??**) .
- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

2.3.1.3 Check Tool Existence

`(Figure Canvas)`

Use this action to check whether a tool entry exists.

Parameters

Name	Data Type	Values	Default
Textpath	String	–	none

Use this paramter to enter the textpath to the tool you want to check.

- Enter the path to the item as a textpath.
- Use slash `'/'` as a path separator (to separate parent nodes from child nodes).
- For example, `File/Open` or `Category/Horror` (without quotes).
- Either make sure that your path is written exactly as it appears in the interface, or use a regular expression to match the text.
- Each segment of the path will be used to find a corresponding node, using the operator provided.

Name	Data Type	Values	Default
Operator	String	equals not equals matches simple match	equals

- The *operator* parameter has four possible values.
- "*not equals*" looks for something that does *not exactly* match.
- "*equals*" looks for an *exact* match.
- Select "*simple match*" to use a simple match expression (→ page **??**) .
- Select "*matches*" to use a regular expression (→ page **??**) .
- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

Name	Data Type	Values	Default
Exists	Boolean	true false	true

- Enter *true* if you expect the tool to exist.
- Enter *false* if you expect the tool *not* to exist.

2.3.1.4 Click Connection

(Figure Canvas)

Use this action to click a connection between two figures.

Parameters

Name	Data Type	Values	Default
Source Text Path	String	–	none

Use this parameter to enter the textpath of the first figure that has this connection.

- Enter the path to the item as a textpath.
- Use slash '/' as a path separator (to separate parent nodes from child nodes).
- For example, File/Open or Category/Horror (without quotes).
- Either make sure that your path is written exactly as it appears in the interface, or use a regular expression to match the text.
- Each segment of the path will be used to find a corresponding node, using the operator provided.

Use the GEF Inspector View to find out the textpath of items on the canvas (→*User Manual* p. 277).

Name	Data Type	Values	Default
Source Operator	String	equals not equals matches simple match	equals

- The *operator* parameter has four possible values.
- "*not equals*" looks for something that does *not exactly* match.
- "*equals*" looks for an *exact* match.
- Select "*simple match*" to use a simple match expression (→ page **??**) .
- Select "*matches*" to use a regular expression (→ page **??**) .
- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

Name	Data Type	Values	Default
Target Text Path	String	–	none

Use this parameter to specify the textpath of the second figure which has this connection.

- Enter the path to the item as a textpath.
- Use slash '/' as a path separator (to separate parent nodes from child nodes).
- For example, File/Open or Category/Horror (without quotes).

- Either make sure that your path is written exactly as it appears in the interface, or use a regular expression to match the text.
- Each segment of the path will be used to find a corresponding node, using the operator provided.

Use the GEF Inspector View to find out the textpath of items on the canvas (→*User Manual* p. 277).

Name	Data Type	Values	Default
Target Operator	String	equals not equals matches simple match	equals

- The *operator* parameter has four possible values.
- "*not equals*" looks for something that does *not exactly* match.
- "*equals*" looks for an *exact* match.
- Select "*simple match*" to use a simple match expression (→ page **??**) .
- Select "*matches*" to use a regular expression (→ page **??**) .
- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

Name	Data Type	Values	Default
Number of Clicks	Integer	–	1

- Enter the amount of clicks that should be performed on the component or item.
- If you enter `0`, the mouse pointer will just be moved over the specified component or item.
- Entering `1` (without quotes) will single-click on the component or item.
- Entering `2` (without quotes) will double-click on the component or item.

Name	Data Type	Values	Default
Mouse Button	Integer	1 2 3	1

- Use this parameter to select which mouse button you want to click with.
  - `1` = left mouse button
  - `2` = middle mouse button
  - `3` = right mouse button
- The mouse button you use to click with can also have an effect on what happens.

- The button to click will depend on your AUT and what the click should achieve.

Do not enter quotes around the mouse button numbers, en them in plain text.

2.3.1.5 Click Figure

(Figure Canvas)

Use this action to click a figure you specify on the canvas.

Parameters

Name	Data Type	Values	Default
Textpath	String	–	none

Use this parameter to specify the textpath to the figure you want to click.

- Enter the path to the item as a textpath.
- Use slash `'/'` as a path separator (to separate parent nodes from child nodes).
- For example, `File/Open` or `Category/Horror` (without quotes).
- Either make sure that your path is written exactly as it appears in the interface, or use a regular expression to match the text.
- Each segment of the path will be used to find a corresponding node, using the operator provided.

Name	Data Type	Values	Default
Operator	String	equals not equals matches simple match	equals

- The *operator* parameter has four possible values.
- "*not equals*" looks for something that does *not exactly* match.
- "*equals*" looks for an *exact* match.
- Select "*simple match*" to use a simple match expression (→ page **??**) .
- Select "*matches*" to use a regular expression (→ page **??**) .
- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

Name	Data Type	Values	Default
Number of Clicks	Integer	–	1

- Enter the amount of clicks that should be performed on the component or item.
- If you enter `0`, the mouse pointer will just be moved over the specified component or item.
- Entering `1` (without quotes) will single-click on the component or item.
- Entering `2` (without quotes) will double-click on the component or item.

Name	Data Type	Values	Default
Mouse Button	Integer	1 2 3	1

- Use this parameter to select which mouse button you want to click with.
  - `1` = left mouse button
  - `2` = middle mouse button
  - `3` = right mouse button
- The mouse button you use to click with can also have an effect on what happens.
- The button to click will depend on your AUT and what the click should achieve.

Do not enter quotes around the mouse button numbers, enter them in plain text.

2.3.1.6 Click in Figure

`(Figure Canvas)`

Use this action to click within a figure you specify. The place to click in the figure is given with X and Y units.

Parameters

Name	Data Type	Values	Default
Textpath	String	–	none

Use this parameter to specify the textpath to the figure you want to click in.

- Enter the path to the item as a textpath.

- Use slash '/' as a path separator (to separate parent nodes from child nodes).
- For example, File/Open or Category/Horror (without quotes).
- Either make sure that your path is written exactly as it appears in the interface, or use a regular expression to match the text.
- Each segment of the path will be used to find a corresponding node, using the operator provided.

Name	Data Type	Values	Default
Operator	String	equals not equals matches simple match	equals

- The *operator* parameter has four possible values.
- "*not equals*" looks for something that does *not exactly* match.
- "*equals*" looks for an *exact* match.
- Select "*simple match*" to use a simple match expression (→ page **??**) .
- Select "*matches*" to use a regular expression (→ page **??**) .
- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

Name	Data Type	Values	Default
Number of Clicks	Integer	–	1

- Enter the amount of clicks that should be performed on the component or item.
- If you enter 0, the mouse pointer will just be moved over the specified component or item.
- Entering 1 (without quotes) will single-click on the component or item.
- Entering 2 (without quotes) will double-click on the component or item.

Name	Data Type	Values	Default
Mouse Button	Integer	1 2 3	1

- Use this parameter to select which mouse button you want to click with.
  - 1 = left mouse button
  - 2 = middle mouse button
  - 3 = right mouse button

- The mouse button you use to click with can also have an effect on what happens.
- The button to click will depend on your AUT and what the click should achieve.

Do not enter quotes around the mouse button numbers, enter them in plain text.

Name	Data Type	Values	Default
x-position	Integer	–	50

- With this parameter, you can specify where in the component you should click.
- Use this parameter to enter a value for the x-axis (the horizontal position).
- Using the x-units parameter, you can specify whether the value should be in pixels or in percent.
- For example, if you enter `50` and then `percent` (without quotes) , then the click will be performed in the middle of the x-axis.
- All positions are calculated from the upper left of the component, with the positive x-axis stretching toward the right side of the component.

We recommend not using `0` (pixels and percent) or `100`(percent) as positions.

Name	Data Type	Values	Default
x-units	String	percent pixel	percent

- Use this parameter to specify whether you want your values for the x-coordinate to be in pixels or percent.
- If you choose percent for this parameter, and `50` for the *x-position* parameter, the click will always be in the middle of the x-axis.

Name	Data Type	Values	Default
y-position	Integer	–	50

- With this parameter, you can specify where in the component you should click.
- Use this parameter to enter a value for the y-axis (the vertical position).
- Using the y-units parameter, you can specify whether the value should be in pixels or in percent.

- For example, if you enter 50 and then percent (without quotes), then the click will be performed in the middle of the y-axis.
- All positions are calculated from the upper left of the component, with the positive y-axis stretching toward the bottom of the component.

We recommend not using 0 (pixels and percent) or 100(percent) positions.

Name	Data Type	Values	Default
y-units	String	percent pixel	percent

- Use this parameter to specify whether you want your values for the y-coordinate to be in pixels or percent.
- If you choose percent for this parameter, and 50 for the *y-position* parameter, the click will always be in the middle of the y-axis.

2.3.1.7 Drag Figure

(Figure Canvas)

- This action drags the figure you specify.
- The cursor is moved over the figure.
- The mouse button you specify is held, and the figure is dragged.
- Follow this action with a *drop* action

If you want to drag multiple items, use an action to select the items you want to drag before using the drag action.

Parameters

Name	Data Type	Values	Default
Textpath	String	–	none

Use this parameter to specify the textpath to the figure you want to drag.

- Enter the path to the item as a textpath.
- Use slash '/' as a path separator (to separate parent nodes from child nodes).

- For example, `File/Open` or `Category/Horror` (without quotes).
- Either make sure that your path is written exactly as it appears in the interface, or use a regular expression to match the text.
- Each segment of the path will be used to find a corresponding node, using the operator provided.

Use the GEF Inspector View to find out the textpath of items on the canvas (→*User Manual* p. 277).

Name	Data Type	Values	Default
Operator	String	equals not equals matches simple match	equals

- The *operator* parameter has four possible values.
- "*not equals*" looks for something that does *not exactly* match.
- "*equals*" looks for an *exact* match.
- Select "*simple match*" to use a simple match expression (→ page **??**) .
- Select "*matches*" to use a regular expression (→ page **??**) .
- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

Name	Data Type	Values	Default
Mouse Button	Integer	1 2 3	1

- Use this parameter to select which mouse button you want to click with.
  - `1` = left mouse button
  - `2` = middle mouse button
  - `3` = right mouse button
- The mouse button you use to click with can also have an effect on what happens.
- The button to click will depend on your AUT and what the click should achieve.

Do not enter quotes around the mouse button numbers, enter them in plain text.

Name	Data Type	Values	Default
Modifier Keys	String	none shift control alt meta cmd mod	none

- Use this parameter to specify a key or keys to be pressed at the same time as the action is being executed.
- You can use this parameter to specify the following keys, or combinations thereof:

  `shift`

  `control` This is the first modifier for Windows and Linux

  `cmd` This is the first modifier for Mac

  `mod` This is the platform-independent first modifier. It corresponds to »CONTROL« for Windows and Linux, and »CMD« for Mac.

  `meta`

  `alt`
- You can also select "*none*" if you do not need a modifier.
- You can enter multiple modifiers for a key combination by entering a space-separated list of modifiers, e.g. `alt shift`.

Do not use quotes around the modifiers, enter them in plain tex

Name	Data Type	Values	Default
x-position	Integer	–	50

- With this parameter, you can specify where in the component you should click.
- Use this parameter to enter a value for the x-axis (the horizontal position).
- Using the x-units parameter, you can specify whether the value should be in pixels or in percent.
- For example, if you enter `50` and then `percent` (without quotes) , then the click will be performed in the middle of the x-axis.
- All positions are calculated from the upper left of the component, with the positive x-axis stretching toward the right side of the component.

We recommend not using `0` (pixels and percent) or `100`(percent) positions.

Name	Data Type	Values	Default
x-units	String	percent pixel	percent

- Use this parameter to specify whether you want your values for the x-coordinate to be in pixels or percent.
- If you choose percent for this parameter, and `50` for the *x-position* parameter, the click will always be in the middle of the x-axis.

Name	Data Type	Values	Default
y-position	Integer	–	50

- With this parameter, you can specify where in the component you should click.
- Use this parameter to enter a value for the y-axis (the vertical position).
- Using the y-units parameter, you can specify whether the value should be in pixels or in percent.
- For example, if you enter `50` and then `percent` (without quotes), then the click will be performed in the middle of the y-axis.
- All positions are calculated from the upper left of the component, with the positive y-axis stretching toward the bottom of the component.

We recommend not using `0` (pixels and percent) or `100`(percent) as positions.

Name	Data Type	Values	Default
y-units	String	percent pixel	percent

- Use this parameter to specify whether you want your values for the y-coordinate to be in pixels or percent.
- If you choose percent for this parameter, and `50` for the *y-position* parameter, the click will always be in the middle of the y-axis.

2.3.1.8 Drop on Figure

`(Figure Canvas)`

- This action drops a dragged figure onto the figure you specify.
- The cursor is moved over the figure.
- The mouse button is let loose, and the figure is dropped.
- Precede this action with a *drag* action

Parameters

Name	Data Type	Values	Default
Textpath	String	–	none

Use this parameter to specify the textpath to the figure you want to drop onto.

- Enter the path to the item as a textpath.
- Use slash `'/'` as a path separator (to separate parent nodes from child nodes).
- For example, `File/Open` or `Category/Horror` (without quotes).
- Either make sure that your path is written exactly as it appears in the interface, or use a regular expression to match the text.
- Each segment of the path will be used to find a corresponding node, using the operator provided.

Use the GEF Inspector View to find out the textpath of items on the canvas (→*User Manual* p. 277).

Name	Data Type	Values	Default
Operator	String	equals not equals matches simple match	equals

- The *operator* parameter has four possible values.
- "*not equals*" looks for something that does *not exactly* match.
- "*equals*" looks for an *exact* match.
- Select "*simple match*" to use a simple match expression (→ page **??**) .
- Select "*matches*" to use a regular expression (→ page **??**) .
- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

Name	Data Type	Values	Default
x-position	Integer	–	50

- With this parameter, you can specify where in the component you should click.
- Use this parameter to enter a value for the x-axis (the horizontal position).
- Using the x-units parameter, you can specify whether the value should be in pixels or in percent.
- For example, if you enter `50` and then `percent` (without quotes) , then the click will be performed in the middle of the x-axis.
- All positions are calculated from the upper left of the component, with the positive x-axis stretching toward the right side of the component.

We recommend not using `0` (pixels and percent) or `100`(percent) as positions.

Name	Data Type	Values	Default
x-units	String	percent pixel	percent

- Use this parameter to specify whether you want your values for the x-coordinate to be in pixels or percent.
- If you choose percent for this parameter, and `50` for the *x-position* parameter, the click will always be in the middle of the x-axis.

Name	Data Type	Values	Default
y-position	Integer	–	50

- With this parameter, you can specify where in the component you should click.
- Use this parameter to enter a value for the y-axis (the vertical position).
- Using the y-units parameter, you can specify whether the value should be in pixels or in percent.
- For example, if you enter `50` and then `percent` (without quotes), then the click will be performed in the middle of the y-axis.
- All positions are calculated from the upper left of the component, with the positive y-axis stretching toward the bottom of the component.

We recommend not using `0` (pixels and percent) or `100`(percent) as positions.

Name	Data Type	Values	Default
y-units	String	percent pixel	percent

- Use this parameter to specify whether you want your values for the y-coordinate to be in pixels or percent.
- If you choose percent for this parameter, and `50` for the *y-position* parameter, the click will always be in the middle of the y-axis.

Name	Data Type	Values	Default
Delay before drop (milliseconds)	Integer	–	100

- Use this parameter to wait before dropping the item you have dragged.
- This can be useful to give the user interface time to scroll etc.
- Give the time to delay in milliseconds.

2.3.1.9 Select Tool

`(Figure Canvas)`

Use this action to select a tool from the panel in the figure canvas.

Parameters

Name	Data Type	Values	Default
Textpath	String	–	none

Use this parameter to specify the textpath to the tool you want to select.

- Enter the path to the item as a textpath.
- Use slash `'/'` as a path separator (to separate parent nodes from child nodes).
- For example, `File/Open` or `Category/Horror` (without quotes).
- Either make sure that your path is written exactly as it appears in the interface, or use a regular expression to match the text.
- Each segment of the path will be used to find a corresponding node, using the operator provided.

Use the GEF Inspector View to find out the textpath of items on the canvas (→*User Manual* p. 277).

Name	Data Type	Values	Default
Operator	String	equals not equals matches simple match	equals

- The *operator* parameter has four possible values.
- "*not equals*" looks for something that does *not exactly* match.
- "*equals*" looks for an *exact* match.
- Select "*simple match*" to use a simple match expression (→ page **??**) .
- Select "*matches*" to use a regular expression (→ page **??**) .
- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

Name	Data Type	Values	Default
Number of Clicks	Integer	–	1

- Enter the amount of clicks that should be performed on the component or item.
- If you enter `0`, the mouse pointer will just be moved over the specified component or item.
- Entering `1` (without quotes) will single-click on the component or item.
- Entering `2` (without quotes) will double-click on the component or item.

2.4 html Toolkit

The HTML toolkit contains components that are specific to HTML applications. You will only have access to these components in your tests if you select HTML as your AUT toolkit.

Unsupported components and actions in the web toolkit
Some components are not supported in web applications because they are either not present or not meaningful in web applications. The unsupported components are:

- Menus
- Tabbed panes
- Trees

2.4.1 (HTML) Anchor

Description:
A HTML hyperlink is a link which can be followed from within a browser.

Synopsis:

- (HTML) Anchor (html)
  - Text Component (abstract)
    - Graphics Component (abstract)

New Actions

Name	Type	Parameters
Check URL (→ page 320)	c	String:Text String:Operator

Inherited Actions

Name	Type	Parameters	Inherited from
Check Enablement (→ page 23)	c	Boolean:Enabled	Graphics Component (abstract)
Check Enablement of Context Menu Entry by Indexpath (→ page 24)	c	String:Indexpath Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)
Check Enablement of Context Menu Entry by Indexpath (Specify Position) (→ page 25)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Indexpath Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)
Check Enablement of Context Menu Entry by Textpath (→ page 28)	c	String:Textpath String:Operator Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)

Check Enablement of Context Menu Entry by Textpath (Specify Position) (→ page 29)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Textpath String:Operator Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)
Check Existence (→ page 32)	c	Boolean:Exists	Graphics Component (abstract)
Check Existence of Context Menu Entry by Indexpath (→ page 33)	c	String:Indexpath Boolean:Exists Integer:Mouse Button	Graphics Component (abstract)
Check Existence of Context Menu Entry by Indexpath (Specify Position) (→ page 34)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Indexpath Boolean:Exists Integer:Mouse Button	Graphics Component (abstract)
Check Existence of Context Menu Entry by Textpath (→ page 37)	c	String:Textpath String:Operator Boolean:Exists Integer:Mouse Button	Graphics Component (abstract)
Check Existence of Context Menu Entry by Textpath (Specify Position) (→ page 38)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Textpath String:Operator Boolean:Exists Integer:Mouse Button	Graphics Component (abstract)

Check Focus (→ page 41)	c	Boolean:Has Focus	Graphics Component (abstract)
Check Property (→ page 42)	c	String:Property Name String:Property Value String:Operator	Graphics Component (abstract)
Check Selection of Context Menu Entry by Indexpath (→ page 43)	c	String:Indexpath Boolean:Selected Integer:Mouse Button	Graphics Component (abstract)
Check Selection of Context Menu Entry by Indexpath (Specify Position) (→ page 45)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Indexpath Boolean:Selected Integer:Mouse Button	Graphics Component (abstract)
Check Selection of Context Menu Entry by Textpath (→ page 47)	c	String:Textpath String:Operator Boolean:Selected Integer:Mouse Button	Graphics Component (abstract)
Check Selection of Context Menu Entry by Textpath (Specify Position) (→ page 49)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Textpath String:Operator Boolean:Selected Integer:Mouse Button	Graphics Component (abstract)

Check Text (→ page 76)	c	String:Text String:Operator	Text Component (abstract)
Click (→ page 52)	e	Integer:Number of Clicks Integer:Mouse Button	Graphics Component (abstract)
Click in Component (→ page 53)	e	Integer:Number of Clicks Integer:Mouse Button Integer:x-position String:x-units Integer:y-position String:y-units	Graphics Component (abstract)
Drag (→ page 55)	e	Integer:Mouse Button String:Modifier Keys Integer:x-position String:x-units Integer:y-position String:y-units	Graphics Component (abstract)
Drop (→ page 58)	e	Integer:x-position String:x-units Integer:y-position String:y-units Integer:Delay before drop (milliseconds)	Graphics Component (abstract)

Select Context Menu Entry by Indexpath (→ page 60)	e	String:Indexpath Integer:Mouse Button	Graphics Component (abstract)
Select Context Menu Entry by Indexpath (Specify Position) (→ page 61)	e	Integer:x-position String:x-units Integer:y-position String:y-units String:Indexpath Integer:Mouse Button	Graphics Component (abstract)
Select Context Menu Entry by Textpath (→ page 64)	e	String:Textpath String:Operator Integer:Mouse Button	Graphics Component (abstract)
Select Context Menu Entry by Textpath (Specify Position) (→ page 66)	e	Integer:x-position String:x-units Integer:y-position String:y-units String:Menupath String:Operator Integer:Mouse Button	Graphics Component (abstract)
Show Text (→ page 69)	e	String:Text Integer:Text Size (in points) Integer:Time per Word (in milliseconds) Integer:Window Width (in pixels)	Graphics Component (abstract)
Store Property (→ page 70)	e	Variable:Variable Name String:Property Name	Graphics Component (abstract)
Store Value (→ page 77)	e	Variable:Variable Name	Text Component (abstract)

Wait for Component (→ page 71)	e	Integer:Timeout in ms Integer:Delay after Visibility	Graphics Component (abstract)

2.4.1.1 Check URL

`((HTML) Anchor)`

- Use this action to check the URL of a link.

Parameters

Name	Data Type	Values	Default
Text	String	–	none

- Use this parameter to enter the URL you expect.
- In Firefox, use the URL as given in the source for the page.

This may not necessarily be the same as the link shown on hovering.

- In IE, use the full URL.
- We recommend entering this parameter between single quotes `'http://www.bredex.de'` to avoid any problems with special characters.

Name	Data Type	Values	Default
Operator	String	equals not equals matches simple match	equals

- The *operator* parameter has four possible values.
- "*not equals*" looks for something that does *not exactly* match.
- "*equals*" looks for an *exact* match.
- Select "*simple match*" to use a simple match expression (→ page **??**) .
- Select "*matches*" to use a regular expression (→ page **??**) .
- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

2.4.2 Browser

Description:

- The *Web Browser* component is the container for all other components in a HTML AUT.
- It represents the AUT as a whole.

Synopsis:

- Browser (html)

New Actions

Name	Type	Parameters
Go Back (→ page 321)	e	
Open URL (→ page 322)	e	String:URL
Reload page (→ page 322)	e	
Select window with identifier (→ page 322)	e	String:Identifier String:Identifier Type
Select window with title (→ page 323)	e	String:Window title String:Operator

Inherited Actions

none

2.4.2.1 Go Back

```
(Browser)
```

- Use this action to go back one page in the browser.
- Use this action instead of actually clicking the back button in the AUT.

This action does not work in Internet Explorer

This action has no parameters.

2.4.2.2 Open URL

`(Browser)`

- Use this action to specify a URL which you would like to open in the current window of the browser.

Parameters

Name	Data Type	Values	Default
URL	String	–	none

- Enter the URL you wish to open, e.g. `http://www.bredex.de`.
- We recommend writing single quotes around the URL so that you do not have to escape special characters: `'http://www.bredex.de'`.

2.4.2.3 Reload page

`(Browser)`

- Use this action to go reload the current page in the browser.
- Use this action instead of actually clicking the refresh button in the AUT.

This action has no parameters.

2.4.2.4 Select window with identifier

`(Browser)`

Use this action to switch to a new window that your HTML AUT has opened during the test, or to switch back to the main window once you have performed steps in a second window. In order to allow this, you must have specified the AUT as supporting multi-window mode in the AUT configuration.

This action allows you to use the window title or other identifiers to select the window.

Parameters

Name	Data Type	Values	Default
Identifier	String	–	none

- Enter the identifier (title, name, variable) of the window that you would like to switch to in your test.

Name	Data Type	Values	Default
Identifier Type	String	title name any	none

Choose the type of identifier you are using.

title If you use the title, this is the window title that you can see at the top of the window. This action only supports *equals* as a matching operator – if you want to be able to match window titles using regular expressions, use the action *Select window with title*.

name You can enter the internal javascript name for the window, if it has one.

any Use this option to be able to enter a javascript variable for the window. You must enter the variable name and the value, as described at:
`http://selenium.googlecode.com/svn/trunk/docs/api/java/com/thoughtworks/selenium/Selenium.html#selectWindow%28java.lang.String%29`
.

If you enter the text *null* for any of these options, the main window will be selected.

2.4.2.5 Select window with title

`(Browser)`

Use this action to switch to a new window that your HTML AUT has opened during the test, or to switch back to the main window once you have performed steps in a second window. In order to allow this, you must have specified the AUT as supporting multi-window mode in the AUT configuration.

This action allows you to use the window title to select the window. In most cases, this should be sufficient. You can, however, use the action *Select wndow with identifier* to be able to use other identifiers.

Parameters

Name	Data Type	Values	Default
Window title	String	–	none

- Enter the title of the window that you would like to switch to in your test.

Name	Data Type	Values	Default
Operator	String	equals not equals matches simple match	equals

- The *operator* parameter has four possible values.
- "*not equals*" looks for something that does *not exactly* match.
- "*equals*" looks for an *exact* match.
- Select "*simple match*" to use a simple match expression (→ page **??**) .
- Select "*matches*" to use a regular expression (→ page **??**) .
- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

2.5 rcp Toolkit

The RCP toolkit contains components that are only found in SWT applications. You will only have access to these components in your tests if you choose RCP or SWT as the Project toolkit.

Most of the RCP components are covered in the section on SWT. Components in GEF AUT's are described in the section on GEF.

2.6 swing Toolkit

The Swing toolkit contains components that can be used on Swing applications. All the Swing components are also concrete components and can be found in the section on concrete components.

2.7 swt Toolkit

The SWT toolkit contains components specific to SWT applications. You will only have access to these components if you select SWT as the toolkit for your AUT.

2.7.1 (SWT) Tool Item

Description:

Toolbar items are buttons and their menus on a toolbar, such as the one in the ITE (Figure 2.33 → page 325). Each of the buttons on the toolbar is a toolbar item.

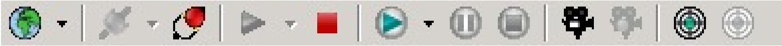


Figure 2.33: *Toolbar items*

Often, toolbar items have drop-down menus (Figure 2.34 → page 325).

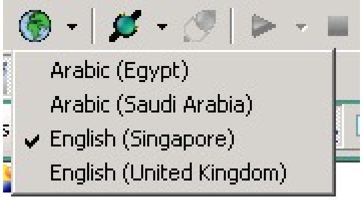


Figure 2.34: *Toolbar item menu*

Mapping toolbar items

In the Object Mapping Mode, a toolbar item to be mapped looks like this:

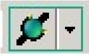


Figure 2.35: *Mapping toolbar items*

Synopsis:

- (SWT) Tool Item (swt)
  - Button Component (concrete)
    - Button Component (abstract)
      - Text Component (abstract)
        Graphics Component (abstract)

New Actions

Name	Type	Parameters
Check Enablement of Entry by Indexpath (→ page 331)	c	String:Indexpath Boolean:Enabled

Check Enablement of Entry by Textpath (→ page 332)	c	String:Menupath String:Operator Boolean:Enabled
Check Existence of Entry by Indexpath (→ page 333)	c	String:Indexpath Boolean:Exists
Check Existence of Entry by Textpath (→ page 334)	c	String:Menupath String:Operator Boolean:Exists
Check Selection of Entry by Indexpath (→ page 335)	c	String:Indexpath Boolean:Selected
Check Selection of Entry by Textpath (→ page 336)	c	String:Menupath String:Operator Boolean:Selected
Select Menu Entry by Indexpath (→ page 337)	e	String:Indexpath
Select Menu Entry by Textpath (→ page 338)	e	String:Menupath String:Operator

Inherited Actions

Name	Type	Parameters	Inherited from
Check Enablement (→ page 23)	c	Boolean:Enabled	Graphics Component (abstract)
Check Enablement of Context Menu Entry by Indexpath (→ page 24)	c	String:Indexpath Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)
Check Enablement of Context Menu Entry by Indexpath (Specify Position) (→ page 25)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Indexpath Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)

Check Enablement of Context Menu Entry by Textpath (→ page 28)	c	String:Textpath String:Operator Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)
Check Enablement of Context Menu Entry by Textpath (Specify Position) (→ page 29)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Textpath String:Operator Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)
Check Existence (→ page 32)	c	Boolean:Exists	Graphics Component (abstract)
Check Existence of Context Menu Entry by Indexpath (→ page 33)	c	String:Indexpath Boolean:Exists Integer:Mouse Button	Graphics Component (abstract)
Check Existence of Context Menu Entry by Indexpath (Specify Position) (→ page 34)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Indexpath Boolean:Exists Integer:Mouse Button	Graphics Component (abstract)
Check Existence of Context Menu Entry by Textpath (→ page 37)	c	String:Textpath String:Operator Boolean:Exists Integer:Mouse Button	Graphics Component (abstract)
Check Existence of Context Menu Entry by Textpath (Specify Position) (→ page 38)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Textpath String:Operator Boolean:Exists Integer:Mouse Button	Graphics Component (abstract)

Check Focus (→ page 41)	c	Boolean:Has Focus	Graphics Component (abstract)
Check Property (→ page 42)	c	String:Property Name String:Property Value String:Operator	Graphics Component (abstract)
Check Selection (→ page 19)	c	Boolean:Selected	Button Component (abstract)
Check Selection of Context Menu Entry by Indexpath (→ page 43)	c	String:Indexpath Boolean:Selected Integer:Mouse Button	Graphics Component (abstract)
Check Selection of Context Menu Entry by Indexpath (Specify Position) (→ page 45)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Indexpath Boolean:Selected Integer:Mouse Button	Graphics Component (abstract)
Check Selection of Context Menu Entry by Textpath (→ page 47)	c	String:Textpath String:Operator Boolean:Selected Integer:Mouse Button	Graphics Component (abstract)
Check Selection of Context Menu Entry by Textpath (Specify Position) (→ page 49)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Textpath String:Operator Boolean:Selected Integer:Mouse Button	Graphics Component (abstract)

Check Text (→ page 76)	c	String:Text String:Operator	Text Component (abstract)
Click (→ page 52)	e	Integer:Number of Clicks Integer:Mouse Button	Graphics Component (abstract)
Click in Component (→ page 53)	e	Integer:Number of Clicks Integer:Mouse Button Integer:x-position String:x-units Integer:y-position String:y-units	Graphics Component (abstract)
Drag (→ page 55)	e	Integer:Mouse Button String:Modifier Keys Integer:x-position String:x-units Integer:y-position String:y-units	Graphics Component (abstract)
Drop (→ page 58)	e	Integer:x-position String:x-units Integer:y-position String:y-units Integer:Delay before drop (milliseconds)	Graphics Component (abstract)

Select Context Menu Entry by Indexpath (→ page 60)	e	String:Indexpath Integer:Mouse Button	Graphics Component (abstract)
Select Context Menu Entry by Indexpath (Specify Position) (→ page 61)	e	Integer:x-position String:x-units Integer:y-position String:y-units String:Indexpath Integer:Mouse Button	Graphics Component (abstract)
Select Context Menu Entry by Textpath (→ page 64)	e	String:Textpath String:Operator Integer:Mouse Button	Graphics Component (abstract)
Select Context Menu Entry by Textpath (Specify Position) (→ page 66)	e	Integer:x-position String:x-units Integer:y-position String:y-units String:Menupath String:Operator Integer:Mouse Button	Graphics Component (abstract)
Show Text (→ page 69)	e	String:Text Integer:Text Size (in points) Integer:Time per Word (in milliseconds) Integer:Window Width (in pixels)	Graphics Component (abstract)
Store Property (→ page 70)	e	Variable:Variable Name String:Property Name	Graphics Component (abstract)
Store Value (→ page 77)	e	Variable:Variable Name	Text Component (abstract)

Wait for Component (→ page 71)	e	Integer:Timeout in ms Integer:Delay after Visibility	Graphics Component (abstract)

2.7.1.1 Check Enablement of Entry by Indexpath

`((SWT) Tool Item)`

- Use this action to check the enablement status of an item in the menu belonging to the toolbar item.
- The item to check is given using the indexpath to the item.

Parameters

Name	Data Type	Values	Default
Indexpath	String	–	none

Use this parameter to specify the menupath to the item you want to check. Make sure you give the whole menupath (start from the very beginning of the menu).

- Enter the path to the item as an indexpath.
- Use slash `'/'` as a path separator (to separate parent nodes from child nodes).
- For example, `1/2` (without quotes).

The first node is '1' (without quotes)

Name	Data Type	Values	Default
Enabled	Boolean	true false	true

- Set this parameter to true if you expect the menu item to be enabled.
- Set the parameter to false if you expect the menu item to be disabled.

2.7.1.2 Check Enablement of Entry by Textpath

`((SWT) Tool Item)`

- Use this action to check whether an item in a menu belonging to a toolbar item is enabled.
- The item to check is given using the textpath to the item.

Parameters

Name	Data Type	Values	Default
Menupath	String	–	none

Use this parameter to specify the menupath to the item you want to check. Make sure you enter the whole path.

- Enter the path to the item as a textpath.
- Use slash `'/'` as a path separator (to separate parent nodes from child nodes).
- For example, `File/Open` or `Category/Horror` (without quotes).
- Either make sure that your path is written exactly as it appears in the interface, or use a regular expression to match the text.
- Each segment of the path will be used to find a corresponding node, using the operator provided.

When you use a regular expression for a textpath, bear in mind that each subpath is considered separately. The first path which corresponds to each subpath will be chosen. It is therefore advisable to make your regular expressions as unambiguous as possible.

Name	Data Type	Values	Default
Operator	String	equals not equals matches simple match	equals

- The *operator* parameter has four possible values.
- "*not equals*" looks for something that does *not exactly* match.
- "*equals*" looks for an *exact* match.
- Select "*simple match*" to use a simple match expression (→ page **??**) .
- Select "*matches*" to use a regular expression (→ page **??**) .
- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

Name	Data Type	Values	Default
Enabled	Boolean	true false	true

- Set this parameter to true if you expect the menu item to be enabled.
- Set the parameter to false if you expect the menu item to be disabled.

2.7.1.3 Check Existence of Entry by Indexpath

`((SWT) Tool Item)`

- Use this action to check the existence of an item in a menu belonging to a toolbar item.
- The item to check is given using the indexpath to the item.

Parameters

Name	Data Type	Values	Default
Indexpath	String	–	none

Use this parameter to specify the menupath to the item you want to check. Make sure you give the whole menupath (start from the very beginning of the menu).

- Enter the path to the item as an indexpath.
- Use slash `'/'` as a path separator (to separate parent nodes from child nodes).
- For example, `1/2` (without quotes).

The first node is '1' (without quotes)

Name	Data Type	Values	Default
Exists	Boolean	true false	true

- Set this parameter to true if you expect the menu item to exist.
- Set the parameter to false if you expect the menu item to *not* exist.

2.7.1.4 Check Existence of Entry by Textpath

`((SWT) Tool Item)`

- Use this action to check the existence of an item in a menu belonging to a toolbar item.
- The item to check is given using the textpath to the item.

Parameters

Name	Data Type	Values	Default
Menupath	String	–	none

Use this parameter to specify the menupath to the item you want to check. Make sure you enter the whole path.

- Enter the path to the item as a textpath.
- Use slash `'/'` as a path separator (to separate parent nodes from child nodes).
- For example, `File/Open` or `Category/Horror` (without quotes).
- Either make sure that your path is written exactly as it appears in the interface, or use a regular expression to match the text.
- Each segment of the path will be used to find a corresponding node, using the operator provided.

When you use a regular expression for a textpath, bear in mind that each subpath is considered separately. The first path which corresponds to each subpath will be chosen. It is therefore advisable to make your regular expressions as unambiguous as possible.

Name	Data Type	Values	Default
Operator	String	equals not equals matches simple match	equals

- The *operator* parameter has four possible values.
- "*not equals*" looks for something that does *not exactly* match.
- "*equals*" looks for an *exact* match.
- Select "*simple match*" to use a simple match expression (→ page **??**) .
- Select "*matches*" to use a regular expression (→ page **??**) .
- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

Name	Data Type	Values	Default
Exists	Boolean	true false	true

- Set this parameter to true if you expect the menu item to exist.
- Set the parameter to false if you expect the menu item to *not* exist.

2.7.1.5 Check Selection of Entry by Indexpath

`((SWT) Tool Item)`

- Use this action to check whether an item in a menu for a toolbar item is selected.
- You enter the menupath to the item, and whether you expect it to be selected or not.
- The menupath is given as an indexpath.

Parameters

Name	Data Type	Values	Default
Indexpath	String	–	none

Use this parameter to specify the menupath to the item you want to check. Make sure you give the whole menupath (start from the very beginning of the menu).

- Enter the path to the item as an indexpath.
- Use slash `'/'` as a path separator (to separate parent nodes from child nodes).
- For example, `1/2` (without quotes).

The first node is '1' (without quotes)

Name	Data Type	Values	Default
Selected	Boolean	true false	true

- Set this parameter to true if you expect the menu item to be selected.
- Set this parameter to false if you expect the menu item *not* to be selected.

2.7.1.6 Check Selection of Entry by Textpath

`((SWT) Tool Item)`

- Use this action to check whether an item in a menu belonging to a toolbar item is selected.
- You enter the menupath to the item, and whether you expect it to be selected or not.
- The menupath is given as a textpath.

Parameters

Name	Data Type	Values	Default
Menupath	String	–	none

Use this parameter to specify the menupath to the item you want to check. Make sure you enter the whole path.

- Enter the path to the item as a textpath.
- Use slash `'/'` as a path separator (to separate parent nodes from child nodes).
- For example, `File/Open` or `Category/Horror` (without quotes).
- Either make sure that your path is written exactly as it appears in the interface, or use a regular expression to match the text.
- Each segment of the path will be used to find a corresponding node, using the operator provided.

When you use a regular expression for a textpath, bear in mind that each subpath is considered separately. The first path which corresponds to each subpath will be chosen. It is therefore advisable to make your regular expressions as unambiguous as possible.

Name	Data Type	Values	Default
Operator	String	equals not equals matches simple match	equals

- The *operator* parameter has four possible values.
- "*not equals*" looks for something that does *not exactly* match.
- "*equals*" looks for an *exact* match.
- Select "*simple match*" to use a simple match expression (→ page **??**) .
- Select "*matches*" to use a regular expression (→ page **??**) .

- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

Name	Data Type	Values	Default
Selected	Boolean	true false	true

- Set this parameter to true if you expect the menu item to be selected.
- Set this parameter to false if you expect the menu item *not* to be selected.

2.7.1.7 Select Menu Entry by Indexpath

`((SWT) Tool Item)`

- Use this action to select an item from a menu belonging to a toolbar item.
- The item you want to select is given using the indexpath to the item.

Parameters

Name	Data Type	Values	Default
Indexpath	String	–	none

Use this parameter to specify the menupath to the item you want to select. Make sure you give the whole menupath (start from the very beginning of the menu).

- Enter the path to the item as an indexpath.
- Use slash `'/'` as a path separator (to separate parent nodes from child nodes).
- For example, `1/2` (without quotes).

The first node is '1' (without quotes)

2.7.1.8 Select Menu Entry by Textpath

`((SWT) Tool Item)`

- Use this action to select an item from a menu belonging to a toolbar item.
- The item you want to select is given using the textpath to the item.

Parameters

Name	Data Type	Values	Default
Menupath	String	–	none

Use this parameter to specify the menupath to the item you want to select. Make sure you enter the whole path.

- Enter the path to the item as a textpath.
- Use slash `'/'` as a path separator (to separate parent nodes from child nodes).
- For example, `File/Open` or `Category/Horror` (without quotes).
- Either make sure that your path is written exactly as it appears in the interface, or use a regular expression to match the text.
- Each segment of the path will be used to find a corresponding node, using the operator provided.

When you use a regular expression for a textpath, bear in mind that each subpath is considered separately. The first path which corresponds to each subpath will be chosen. It is therefore advisable to make your regular expressions as unambiguous as possible.

Name	Data Type	Values	Default
Operator	String	equals not equals matches simple match	equals

- The *operator* parameter has four possible values.
- "*not equals*" looks for something that does *not exactly* match.
- "*equals*" looks for an *exact* match.
- Select "*simple match*" to use a simple match expression (→ page **??**) .
- Select "*matches*" to use a regular expression (→ page **??**) .
- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

2.7.2 (SWT) Tree

Description:
A tree in SWT is essentially the same as a tree in Swing, with an extra option to have chackboxes on the nodes of the tree. Use this component if you want to select or check the selection of the checkboxes on an SWT tree. For other actions on trees, use the tree component in the *concrete* toolkit.

Synopsis:

- (SWT) Tree (swt)
  - Tree Component (concrete)
    - ⋆ Graphics Component (abstract)

New Actions

Name	Type	Parameters
Check Selection of Checkbox by Indexpath (→ page 346)	c	String:Path Type Integer:Pre-ascend String:Indexpath Boolean:Checked
Check Selection of Checkbox by Textpath (→ page 348)	c	String:Path Type Integer:Pre-ascend String:Textpath String:Operator Boolean:Checked
Check Selection of Checkbox on Selected Node (→ page 350)	c	Boolean:Checked
Toggle Checkbox on Node by Indexpath (→ page 350)	e	String:Path Type Integer:Pre-ascend String:Indexpath
Toggle Checkbox on Node by Textpath (→ page 351)	e	String:Path Type Integer:Pre-ascend String:Textpath String:Operator

Inherited Actions

Name	Type	Parameters	Inherited from
Check Enablement (→ page 23)	c	Boolean:Enabled	Graphics Component (abstract)
Check Enablement of Context Menu Entry by Indexpath (→ page 24)	c	String:Indexpath Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)

Check Enablement of Context Menu Entry by Indexpath (Specify Position) (→ page 25)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Indexpath Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)
Check Enablement of Context Menu Entry by Textpath (→ page 28)	c	String:Textpath String:Operator Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)
Check Enablement of Context Menu Entry by Textpath (Specify Position) (→ page 29)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Textpath String:Operator Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)
Check Existence (→ page 32)	c	Boolean:Exists	Graphics Component (abstract)
Check Existence of Context Menu Entry by Indexpath (→ page 33)	c	String:Indexpath Boolean:Exists Integer:Mouse Button	Graphics Component (abstract)
Check Existence of Context Menu Entry by Indexpath (Specify Position) (→ page 34)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Indexpath Boolean:Exists Integer:Mouse Button	Graphics Component (abstract)

Check Existence of Context Menu Entry by Textpath (→ page 37)	c	String:Textpath String:Operator Boolean:Exists Integer:Mouse Button	Graphics Component (abstract)
Check Existence of Context Menu Entry by Textpath (Specify Position) (→ page 38)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Textpath String:Operator Boolean:Exists Integer:Mouse Button	Graphics Component (abstract)
Check Existence of Node by Indexpath (→ page 257)	c	String:Path Type Integer:Pre-ascend String:Indexpath Boolean:Exists	Tree Component (concrete)
Check Existence of Node by Textpath (→ page 258)	c	String:Path Type Integer:Pre-ascend String:Textpath String:Operator Boolean:Exists	Tree Component (concrete)
Check Focus (→ page 41)	c	Boolean:Has Focus	Graphics Component (abstract)
Check Property (→ page 42)	c	String:Property Name String:Property Value String:Operator	Graphics Component (abstract)
Check Selection of Context Menu Entry by Indexpath (→ page 43)	c	String:Indexpath Boolean:Selected Integer:Mouse Button	Graphics Component (abstract)

Check Selection of Context Menu Entry by Indexpath (Specify Position) (→ page 45)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Indexpath Boolean:Selected Integer:Mouse Button	Graphics Component (abstract)
Check Selection of Context Menu Entry by Textpath (→ page 47)	c	String:Textpath String:Operator Boolean:Selected Integer:Mouse Button	Graphics Component (abstract)
Check Selection of Context Menu Entry by Textpath (Specify Position) (→ page 49)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Textpath String:Operator Boolean:Selected Integer:Mouse Button	Graphics Component (abstract)
Check Text (Mouse Position) (→ page 260)	c	String:Text String:Operator	Tree Component (concrete)
Check Text of Selected Node(s) (→ page 261)	c	String:Text String:Operator	Tree Component (concrete)
Click (→ page 52)	e	Integer:Number of Clicks Integer:Mouse Button	Graphics Component (abstract)
Click in Component (→ page 53)	e	Integer:Number of Clicks Integer:Mouse Button Integer:x-position String:x-units Integer:y-position String:y-units	Graphics Component (abstract)

Collapse Node by Indexpath (→ page 262)	e	String:Path Type Integer:Pre-ascend String:Indexpath	Tree Component (concrete)
Collapse Node by Textpath (→ page 264)	e	String:Path Type Integer:Pre-ascend String:Textpath String:Operator	Tree Component (concrete)
Drag (→ page 55)	e	Integer:Mouse Button String:Modifier Keys Integer:x-position String:x-units Integer:y-position String:y-units	Graphics Component (abstract)
Drag Node by Indexpath (→ page 267)	e	Integer:Mouse Button String:Modifier Keys String:Path Type Integer:Pre-ascend String:Tree Indexpath	Tree Component (concrete)
Drag Node by Textpath (→ page 269)	e	Integer:Mouse Button String:Modifier Keys String:Path Type Integer:Pre-ascend String:Tree Textpath String:Operator	Tree Component (concrete)

Drop (→ page 58)	e	Integer:x-position String:x-units Integer:y-position String:y-units Integer:Delay before drop (milliseconds)	Graphics Component (abstract)
Drop on Node by Indexpath (→ page 272)	e	String:Path Type Integer:Pre-ascend String:Tree Indexpath Integer:Delay before drop (milliseconds)	Tree Component (concrete)
Drop on Node by Textpath (→ page 274)	e	String:Path Type Integer:Pre-ascend String:Tree Textpath String:Operator Integer:Delay before drop (milliseconds)	Tree Component (concrete)
Expand Node by Indexpath (→ page 276)	e	String:Path Type Integer:Pre-ascend String:Indexpath	Tree Component (concrete)
Expand Node by Textpath (→ page 277)	e	String:Path Type Integer:Pre-ascend String:Textpath String:Operator	Tree Component (concrete)
Move (→ page 280)	e	String:Direction Integer:Number of Nodes Integer:Number of Clicks	Tree Component (concrete)

Select Context Menu Entry by Indexpath (→ page 60)	e	String:Indexpath Integer:Mouse Button	Graphics Component (abstract)
Select Context Menu Entry by Indexpath (Specify Position) (→ page 61)	e	Integer:x-position String:x-units Integer:y-position String:y-units String:Indexpath Integer:Mouse Button	Graphics Component (abstract)
Select Context Menu Entry by Textpath (→ page 64)	e	String:Textpath String:Operator Integer:Mouse Button	Graphics Component (abstract)
Select Context Menu Entry by Textpath (Specify Position) (→ page 66)	e	Integer:x-position String:x-units Integer:y-position String:y-units String:Menupath String:Operator Integer:Mouse Button	Graphics Component (abstract)
Select Node by Indexpath (→ page 281)	e	String:Path Type Integer:Pre-ascend String:Indexpath Integer:Number of Clicks Integer:Mouse Button String:Extend Selection	Tree Component (concrete)

Select Node by Textpath (→ page 283)	e	String:Path Type Integer:Pre-ascend String:Textpath String:Operator Integer:Number of Clicks Integer:Mouse Button String:Extend Selection	Tree Component (concrete)
Show Text (→ page 69)	e	String:Text Integer:Text Size (in points) Integer:Time per Word (in milliseconds) Integer:Window Width (in pixels)	Graphics Component (abstract)
Store Property (→ page 70)	e	Variable:Variable Name String:Property Name	Graphics Component (abstract)
Store Text at Mouse Position (→ page 286)	e	Variable:Variable Name	Tree Component (concrete)
Store Text of Selected Node (→ page 286)	e	Variable:Variable Name	Tree Component (concrete)
Wait for Component (→ page 71)	e	Integer:Timeout in ms Integer:Delay after Visibility	Graphics Component (abstract)

2.7.2.1 Check Selection of Checkbox by Indexpath

`((SWT) Tree)`

h

- This action checks whether a checkbox on a node in the tree is selected.
- You give the path to the node whose checkbox you want to check as an indexpath.

Parameters

Name	Data Type	Values	Default
Path Type	String	absolute relative	absolute

- The *path type* parameter has two values.
- "*absolute*" begins searching for the path at the root of the tree. Use this value when you enter the full path.
- Select "*relative*" to begin the search for the path at the currently selected node. This allows you to enter a partial path, ignoring all elements above the selected node.

Name	Data Type	Values	Default
Pre-ascend	Integer	–	0

- Use this parameter with a *relative path* to start the search above the currently selected node.
- The search will begin *n* nodes above the currently selected node, where *n* is the value of this parameter.
- For example, a value of 0 will simply begin with the currently selected node, and a value of 2 will begin with the parent of the parent of the currently selected node.
- A value can be given such that the search begins 1 level above the topmost visible level. This is then equivalent to an *absolute path*.
- If the search would begin any higher than this, the action fails: The tree node cannot be found.

The *Pre-ascend* parameter is simply ignored for *absolute paths*.

Name	Data Type	Values	Default
Indexpath	String	–	none

Use this parameter to specify the textpath to the node where the checkbox is whose status you want to check. Make sure you give the whole path (either starting from the top of the tree, or at the position defined by the pre-ascend and path type parameters).

- Enter the path to the item as an indexpath.
- Use slash `'/'` as a path separator (to separate parent nodes from child nodes).
- For example, `1/2` (without quotes).

The first node is '1' (without quotes)

Name	Data Type	Values	Default
Checked	Boolean	true false	true

- Set to true if you expect the checkbox to be selected.
- Set to false if you expect the checkbox *not* to be selected.

2.7.2.2 Check Selection of Checkbox by Textpath

`((SWT) Tree)`

- This action checks whether a checkbox on a node in the tree is selected.
- You give the path to the node whose checkbox you want to check as a textpath.

Parameters

Name	Data Type	Values	Default
Path Type	String	absolute relative	absolute

- The *path type* parameter has two values.
- "*absolute*" begins searching for the path at the root of the tree. Use this value when you enter the full path.
- Select "*relative*" to begin the search for the path at the currently selected node. This allows you to enter a partial path, ignoring all elements above the selected node.

Name	Data Type	Values	Default
Pre-ascend	Integer	–	0

- Use this parameter with a *relative path* to start the search above the currently selected node.
- The search will begin *n* nodes above the currently selected node, where *n* is the value of this parameter.
- For example, a value of 0 will simply begin with the currently selected node, and a value of 2 will begin with the parent of the parent of the currently selected node.
- A value can be given such that the search begins 1 level above the topmost visible level. This is then equivalent to an *absolute path*.
- If the search would begin any higher than this, the action fails: The tree node cannot be found.

The *Pre-ascend* parameter is simply ignored for *absolute paths*.

Name	Data Type	Values	Default
Textpath	String	–	none

Enter the textpath to the node on which you want to check the status of the checkbox. Make sure you give the whole path (either starting from the top of the tree, or at the position defined by the pre-ascend and path type parameters).

- Enter the path to the item as a textpath.
- Use slash `'/'` as a path separator (to separate parent nodes from child nodes).
- For example, `File/Open` or `Category/Horror` (without quotes).
- Either make sure that your path is written exactly as it appears in the interface, or use a regular expression to match the text.
- Each segment of the path will be used to find a corresponding node, using the operator provided.

Name	Data Type	Values	Default
Operator	String	equals not equals matches simple match	equals

- The *operator* parameter has four possible values.
- "*not equals*" looks for something that does *not exactly* match.
- "*equals*" looks for an *exact* match.
- Select "*simple match*" to use a simple match expression (→ page **??**) .
- Select "*matches*" to use a regular expression (→ page **??**) .
- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

When you use a regular expression for a textpath, bear in mind that each subpath is considered separately. The first path which corresponds to each subpath will be chosen. It is therefore advisable to make your regular expressions as unambiguous as possible.

Name	Data Type	Values	Default
Checked	Boolean	true false	true

- Set to true if you expect the checkbox to be selected.
- Set to false if you expect the checkbox *not* to be selected.

2.7.2.3 Check Selection of Checkbox on Selected Node

`((SWT) Tree)`

- Use this action to check whether the checkbox on the currently selected node is selected or not.

Parameters

Name	Data Type	Values	Default
Checked	Boolean	true false	true

- Set to true if you expect the checkbox to be selected.
- Set to false if you expect the checkbox *not* to be selected.

2.7.2.4 Toggle Checkbox on Node by Indexpath

`((SWT) Tree)`

- Use this action to select or deselect a checkbox on a node you specify.
- You give the path to the node as an indexpath.
- If the checkbox is currently selected, this action will deselect it and vice-versa.

Parameters

Name	Data Type	Values	Default
Path Type	String	absolute relative	absolute

- The *path type* parameter has two values.
- "*absolute*" begins searching for the path at the root of the tree. Use this value when you enter the full path.

- Select "*relative*" to begin the search for the path at the currently selected node. This allows you to enter a partial path, ignoring all elements above the selected node.

Name	Data Type	Values	Default
Pre-ascend	Integer	–	0

- Use this parameter with a *relative path* to start the search above the currently selected node.
- The search will begin *n* nodes above the currently selected node, where *n* is the value of this parameter.
- For example, a value of 0 will simply begin with the currently selected node, and a value of 2 will begin with the parent of the parent of the currently selected node.
- A value can be given such that the search begins 1 level above the topmost visible level. This is then equivalent to an *absolute path*.
- If the search would begin any higher than this, the action fails: The tree node cannot be found.

The *Pre-ascend* parameter is simply ignored for *absolute paths*.

Name	Data Type	Values	Default
Indexpath	String	–	none

Use this parameter to specify the textpath to the node you want to select / deselect the checkbox on. Make sure you give the whole path (either starting from the top of the tree, or at the position defined by the pre-ascend and path type parameters).

- Enter the path to the item as an indexpath.
- Use slash `'/'` as a path separator (to separate parent nodes from child nodes).
- For example, `1/2` (without quotes).

The first node is '1' (without quotes)

2.7.2.5 Toggle Checkbox on Node by Textpath

`((SWT) Tree)`

- Use this action to select or deselect a checkbox on a node you specify.

- You give the path to the node as a textpath.
- If the checkbox is currently selected, this action will deselect it and vice-versa.

Parameters

Name	Data Type	Values	Default
Path Type	String	absolute relative	absolute

- The *path type* parameter has two values.
- "*absolute*" begins searching for the path at the root of the tree. Use this value when you enter the full path.
- Select "*relative*" to begin the search for the path at the currently selected node. This allows you to enter a partial path, ignoring all elements above the selected node.

Name	Data Type	Values	Default
Pre-ascend	Integer	–	0

- Use this parameter with a *relative path* to start the search above the currently selected node.
- The search will begin *n* nodes above the currently selected node, where *n* is the value of this parameter.
- For example, a value of 0 will simply begin with the currently selected node, and a value of 2 will begin with the parent of the parent of the currently selected node.
- A value can be given such that the search begins 1 level above the topmost visible level. This is then equivalent to an *absolute path*.
- If the search would begin any higher than this, the action fails: The tree node cannot be found.

The *Pre-ascend* parameter is simply ignored for *absolute paths*.

Name	Data Type	Values	Default
Textpath	String	–	none

Enter the textpath to the node on which you want to select or deselect the checkbox. Make sure you give the whole path (either starting from the top of the tree, or at the position defined by the pre-ascend and path type parameters).

- Enter the path to the item as a textpath.
- Use slash `'/'` as a path separator (to separate parent nodes from child nodes).

- For example, `File/Open` or `Category/Horror` (without quotes).
- Either make sure that your path is written exactly as it appears in the interface, or use a regular expression to match the text.
- Each segment of the path will be used to find a corresponding node, using the operator provided.

Name	Data Type	Values	Default
Operator	String	equals not equals matches simple match	equals

- The *operator* parameter has four possible values.
- "*not equals*" looks for something that does *not exactly* match.
- "*equals*" looks for an *exact* match.
- Select "*simple match*" to use a simple match expression (→ page **??**) .
- Select "*matches*" to use a regular expression (→ page **??**) .
- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

When you use a regular expression for a textpath, bear in mind that each subpath is considered separately. The first path which corresponds to each subpath will be chosen. It is therefore advisable to make your regular expressions as unambiguous as possible.

Used By

(SWT) Tree Table (swt)

2.7.3 (SWT) Tree Table

Description:

A tree table is an SWT component which has some of the proeprties of a tree (e.g. nodes which can be expanded or collapsed) and some of the properties of a table (e.g. rows and columns).

Because the forward slash (/) is a special symbol for trees, if you want to use a slash as part of your parameter value, you have to mask it. See the section later in this document (→ page **??**) for more details.

Synopsis:

- (SWT) Tree Table (swt)
  - (SWT) Tree (swt)
    - Tree Component (concrete)
      - Graphics Component (abstract)

New Actions

Name	Type	Parameters
Check Selection by Value(s) (Specify Column) (→ page 363)	c	String:Text String:Operator Integer:Column
Select Entry by Indexpath (Specify Column) (→ page 364)	e	String:Path Type Integer:Pre-ascend String:Indexpath Integer:Number of Clicks Integer:Column Integer:Mouse Button
Select Entry by Textpath (Specify Column) (→ page 366)	e	String:Path Type Integer:Pre-ascend String:Textpath String:Operator Integer:Number of Clicks Integer:Column Integer:Mouse Button

Inherited Actions

Name	Type	Parameters	Inherited from
Check Enablement (→ page 23)	c	Boolean:Enabled	Graphics Component (abstract)
Check Enablement of Context Menu Entry by Indexpath (→ page 24)	c	String:Indexpath Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)
Check Enablement of Context Menu Entry by Indexpath (Specify Position) (→ page 25)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Indexpath Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)
Check Enablement of Context Menu Entry by Textpath (→ page 28)	c	String:Textpath String:Operator Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)
Check Enablement of Context Menu Entry by Textpath (Specify Position) (→ page 29)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Textpath String:Operator Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)

Check Existence (→ page 32)	c	Boolean:Exists	Graphics Component (abstract)
Check Existence of Context Menu Entry by Indexpath (→ page 33)	c	String:Indexpath Boolean:Exists Integer:Mouse Button	Graphics Component (abstract)
Check Existence of Context Menu Entry by Indexpath (Specify Position) (→ page 34)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Indexpath Boolean:Exists Integer:Mouse Button	Graphics Component (abstract)
Check Existence of Context Menu Entry by Textpath (→ page 37)	c	String:Textpath String:Operator Boolean:Exists Integer:Mouse Button	Graphics Component (abstract)
Check Existence of Context Menu Entry by Textpath (Specify Position) (→ page 38)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Textpath String:Operator Boolean:Exists Integer:Mouse Button	Graphics Component (abstract)
Check Existence of Node by Indexpath (→ page 257)	c	String:Path Type Integer:Pre-ascend String:Indexpath Boolean:Exists	Tree Component (concrete)
Check Existence of Node by Textpath (→ page 258)	c	String:Path Type Integer:Pre-ascend String:Textpath String:Operator Boolean:Exists	Tree Component (concrete)

Check Focus (→ page 41)	c	Boolean:Has Focus	Graphics Component (abstract)
Check Property (→ page 42)	c	String:Property Name String:Property Value String:Operator	Graphics Component (abstract)
Check Selection of Checkbox by Indexpath (→ page 346)	c	String:Path Type Integer:Pre-ascend String:Indexpath Boolean:Checked	(SWT) Tree (swt)
Check Selection of Checkbox by Textpath (→ page 348)	c	String:Path Type Integer:Pre-ascend String:Textpath String:Operator Boolean:Checked	(SWT) Tree (swt)
Check Selection of Checkbox on Selected Node (→ page 350)	c	Boolean:Checked	(SWT) Tree (swt)
Check Selection of Context Menu Entry by Indexpath (→ page 43)	c	String:Indexpath Boolean:Selected Integer:Mouse Button	Graphics Component (abstract)
Check Selection of Context Menu Entry by Indexpath (Specify Position) (→ page 45)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Indexpath Boolean:Selected Integer:Mouse Button	Graphics Component (abstract)

Check Selection of Context Menu Entry by Textpath (→ page 47)	c	String:Textpath String:Operator Boolean:Selected Integer:Mouse Button	Graphics Component (abstract)
Check Selection of Context Menu Entry by Textpath (Specify Position) (→ page 49)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Textpath String:Operator Boolean:Selected Integer:Mouse Button	Graphics Component (abstract)
Check Text (Mouse Position) (→ page 260)	c	String:Text String:Operator	Tree Component (concrete)
Check Text of Selected Node(s) (→ page 261)	c	String:Text String:Operator	Tree Component (concrete)
Click (→ page 52)	e	Integer:Number of Clicks Integer:Mouse Button	Graphics Component (abstract)
Click in Component (→ page 53)	e	Integer:Number of Clicks Integer:Mouse Button Integer:x-position String:x-units Integer:y-position String:y-units	Graphics Component (abstract)

Collapse Node by Indexpath (→ page 262)	e	String:Path Type Integer:Pre-ascend String:Indexpath	Tree Component (concrete)
Collapse Node by Textpath (→ page 264)	e	String:Path Type Integer:Pre-ascend String:Textpath String:Operator	Tree Component (concrete)
Drag (→ page 55)	e	Integer:Mouse Button String:Modifier Keys Integer:x-position String:x-units Integer:y-position String:y-units	Graphics Component (abstract)
Drag Node by Indexpath (→ page 267)	e	Integer:Mouse Button String:Modifier Keys String:Path Type Integer:Pre-ascend String:Tree Indexpath	Tree Component (concrete)
Drag Node by Textpath (→ page 269)	e	Integer:Mouse Button String:Modifier Keys String:Path Type Integer:Pre-ascend String:Tree Textpath String:Operator	Tree Component (concrete)

Drop (→ page 58)	e	Integer:x-position String:x-units Integer:y-position String:y-units Integer:Delay before drop (milliseconds)	Graphics Component (abstract)
Drop on Node by Indexpath (→ page 272)	e	String:Path Type Integer:Pre-ascend String:Tree Indexpath Integer:Delay before drop (milliseconds)	Tree Component (concrete)
Drop on Node by Textpath (→ page 274)	e	String:Path Type Integer:Pre-ascend String:Tree Textpath String:Operator Integer:Delay before drop (milliseconds)	Tree Component (concrete)
Expand Node by Indexpath (→ page 276)	e	String:Path Type Integer:Pre-ascend String:Indexpath	Tree Component (concrete)
Expand Node by Textpath (→ page 277)	e	String:Path Type Integer:Pre-ascend String:Textpath String:Operator	Tree Component (concrete)
Move (→ page 280)	e	String:Direction Integer:Number of Nodes Integer:Number of Clicks	Tree Component (concrete)

Select Context Menu Entry by Indexpath (→ page 60)	e	String:Indexpath Integer:Mouse Button	Graphics Component (abstract)
Select Context Menu Entry by Indexpath (Specify Position) (→ page 61)	e	Integer:x-position String:x-units Integer:y-position String:y-units String:Indexpath Integer:Mouse Button	Graphics Component (abstract)
Select Context Menu Entry by Textpath (→ page 64)	e	String:Textpath String:Operator Integer:Mouse Button	Graphics Component (abstract)
Select Context Menu Entry by Textpath (Specify Position) (→ page 66)	e	Integer:x-position String:x-units Integer:y-position String:y-units String:Menupath String:Operator Integer:Mouse Button	Graphics Component (abstract)
Select Node by Indexpath (→ page 281)	e	String:Path Type Integer:Pre-ascend String:Indexpath Integer:Number of Clicks Integer:Mouse Button String:Extend Selection	Tree Component (concrete)

Select Node by Textpath (→ page 283)	e	String:Path Type Integer:Pre-ascend String:Textpath String:Operator Integer:Number of Clicks Integer:Mouse Button String:Extend Selection	Tree Component (concrete)
Show Text (→ page 69)	e	String:Text Integer:Text Size (in points) Integer:Time per Word (in milliseconds) Integer:Window Width (in pixels)	Graphics Component (abstract)
Store Property (→ page 70)	e	Variable:Variable Name String:Property Name	Graphics Component (abstract)
Store Text at Mouse Position (→ page 286)	e	Variable:Variable Name	Tree Component (concrete)
Store Text of Selected Node (→ page 286)	e	Variable:Variable Name	Tree Component (concrete)
Toggle Checkbox on Node by Indexpath (→ page 350)	e	String:Path Type Integer:Pre-ascend String:Indexpath	(SWT) Tree (swt)
Toggle Checkbox on Node by Textpath (→ page 351)	e	String:Path Type Integer:Pre-ascend String:Textpath String:Operator	(SWT) Tree (swt)
Wait for Component (→ page 71)	e	Integer:Timeout in ms Integer:Delay after Visibility	Graphics Component (abstract)

2.7.3.1 Check Selection by Value(s) (Specify Column)

`((SWT) Tree Table)`

- Use this item to check the text of the currently selected node in the specified column.

  Parameters

Name	Data Type	Values	Default
Text	String	–	none

  - Enter the value you want to test.

Name	Data Type	Values	Default
Operator	String	equals not equals matches simple match	equals

  - The *operator* parameter has four possible values.
  - "*not equals*" looks for something that does *not exactly* match.
  - "*equals*" looks for an *exact* match.
  - Select "*simple match*" to use a simple match expression (→ page **??**) .
  - Select "*matches*" to use a regular expression (→ page **??**) .
  - Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

Name	Data Type	Values	Default
Column	Integer	–	1

  - In tables, you can address the header of a table either by using its index or by entering its title.
  - The first column is `1` (without quotes) and so on. The index `0` (without quotes) refers to the header itself.
  - If the cell you want to execute an action on is in the fourth column and is called *Price*, you can address the column either with the index `4` (without quotes) or with the string *Price*.
  - In JavaFX AUT's, you can also enter a path to the column to test tables that have nested columns (i.e. where a column has child columns). Enter the column as a path, separated with a forward slash, e.g. `Address/Street`. You can use indexpaths and textpaths for columns in JavaFX tables.

2.7.3.2 Select Entry by Indexpath (Specify Column)

`((SWT) Tree Table)`

- Use this item to select a node from a tree table.
- You give the path to the node using an indexpath and by specifying which column to use.
- This is achieved by:
  - Searching for the node and column you specify.
  - If the tree is not expanded, it is expanded.
  - Clicking on the node, the amount of times you specify.

Parameters

Name	Data Type	Values	Default
Path Type	String	absolute relative	absolute

- The *path type* parameter has two values.
- "*absolute*" begins searching for the path at the root of the tree. Use this value when you enter the full path.
- Select "*relative*" to begin the search for the path at the currently selected node. This allows you to enter a partial path, ignoring all elements above the selected node.

Name	Data Type	Values	Default
Pre-ascend	Integer	–	0

- Use this parameter with a *relative path* to start the search above the currently selected node.
- The search will begin *n* nodes above the currently selected node, where *n* is the value of this parameter.
- For example, a value of 0 will simply begin with the currently selected node, and a value of 2 will begin with the parent of the parent of the currently selected node.
- A value can be given such that the search begins 1 level above the topmost visible level. This is then equivalent to an *absolute path*.
- If the search would begin any higher than this, the action fails: The tree node cannot be found.

The *Pre-ascend* parameter is simply ignored for *absolute paths*.

Name	Data Type	Values	Default
Indexpath	String	–	none

Use this parameter to specify the indexpath of the subtree you want to select. Make sure you give the whole path (either starting from the top of the tree, or at the position defined by the pre-ascend and path type parameters).

- Enter the path to the item as an indexpath.
- Use slash `'/'` as a path separator (to separate parent nodes from child nodes).
- For example, `1/2` (without quotes).

The first node is '1' (without quotes)

Name	Data Type	Values	Default
Number of Clicks	Integer	–	1

- Enter the amount of clicks that should be performed on the component or item.
- If you enter `0`, the mouse pointer will just be moved over the specified component or item.
- Entering `1` (without quotes) will single-click on the component or item.
- Entering `2` (without quotes) will double-click on the component or item.

Name	Data Type	Values	Default
Column	Integer	–	1

- In tables, you can address the header of a table either by using its index or by entering its title.
- The first column is `1` (without quotes) and so on. The index `0` (without quotes) refers to the header itself.
- If the cell you want to execute an action on is in the fourth column and is called *Price*, you can address the column either with the index `4` (without quotes) or with the string *Price*.
- In JavaFX AUT's, you can also enter a path to the column to test tables that have nested columns (i.e. where a column has child columns). Enter the column as a path, separated with a forward slash, e.g. `Address/Street`. You can use indexpaths and textpaths for columns in JavaFX tables.

Name	Data Type	Values	Default
Mouse Button	Integer	1 2 3	1

- Use this parameter to select which mouse button you want to click with.
  - `1` = left mouse button
  - `2` = middle mouse button
  - `3` = right mouse button
- The mouse button you use to click with can also have an effect on what happens.
- The button to click will depend on your AUT and what the click should achieve.

Do not enter quotes around the mouse button numbers, enter them in plain text.

2.7.3.3 Select Entry by Textpath (Specify Column)

`((SWT) Tree Table)`

- Use this item to select a node from a tree table.
- You give the path to the node using a textpath and by specifying which column to use.
- This is achieved by:
  - Searching for the node and column you specify.
  - If the tree is not expanded, it is expanded.
  - Clicking on the node, the amount of times you specify.

Parameters

Name	Data Type	Values	Default
Path Type	String	absolute relative	absolute

- The *path type* parameter has two values.
- "*absolute*" begins searching for the path at the root of the tree. Use this value when you enter the full path.
- Select "*relative*" to begin the search for the path at the currently selected node. This allows you to enter a partial path, ignoring all elements above the selected node.

Name	Data Type	Values	Default
Pre-ascend	Integer	–	0

- Use this parameter with a *relative path* to start the search above the currently selected node.
- The search will begin *n* nodes above the currently selected node, where *n* is the value of this parameter.
- For example, a value of 0 will simply begin with the currently selected node, and a value of 2 will begin with the parent of the parent of the currently selected node.
- A value can be given such that the search begins 1 level above the topmost visible level. This is then equivalent to an *absolute path*.
- If the search would begin any higher than this, the action fails: The tree node cannot be found.

The *Pre-ascend* parameter is simply ignored for *absolute paths*.

Name	Data Type	Values	Default
Textpath	String	–	none

Use this parameter to specify the textpath of the subtree you want to select. Make sure you give the whole path (either starting from the top of the tree, or at the position defined by the pre-ascend and path type parameters).

- Enter the path to the item as a textpath.
- Use slash `'/'` as a path separator (to separate parent nodes from child nodes).
- For example, `File/Open` or `Category/Horror` (without quotes).
- Either make sure that your path is written exactly as it appears in the interface, or use a regular expression to match the text.
- Each segment of the path will be used to find a corresponding node, using the operator provided.

When you use a regular expression for a textpath, bear in mind that each subpath is considered separately. The first path which corresponds to each subpath will be chosen. It is therefore advisable to make your regular expressions as unambiguous as possible.

Name	Data Type	Values	Default
Operator	String	equals not equals matches simple match	equals

- The *operator* parameter has four possible values.

- "*not equals*" looks for something that does *not exactly* match.
- "*equals*" looks for an *exact* match.
- Select "*simple match*" to use a simple match expression (→ page **??**) .
- Select "*matches*" to use a regular expression (→ page **??**) .
- Regardless of the operator used: If a selection action is being used and there is more than one match, then the first item found will be chosen.

Name	Data Type	Values	Default
Number of Clicks	Integer	–	1

- Enter the amount of clicks that should be performed on the component or item.
- If you enter `0`, the mouse pointer will just be moved over the specified component or item.
- Entering `1` (without quotes) will single-click on the component or item.
- Entering `2` (without quotes) will double-click on the component or item.

Name	Data Type	Values	Default
Column	Integer	–	1

- In tables, you can address the header of a table either by using its index or by entering its title.
- The first column is `1` (without quotes) and so on. The index `0` (without quotes) refers to the header itself.
- If the cell you want to execute an action on is in the fourth column and is called *Price*, you can address the column either with the index `4` (without quotes) or with the string *Price*.
- In JavaFX AUT's, you can also enter a path to the column to test tables that have nested columns (i.e. where a column has child columns). Enter the column as a path, separated with a forward slash, e.g. `Address/Street`. You can use indexpaths and textpaths for columns in JavaFX tables.

Name	Data Type	Values	Default
Mouse Button	Integer	1 2 3	1

- Use this parameter to select which mouse button you want to click with.
  - `1` = left mouse button
  - `2` = middle mouse button
  - `3` = right mouse button

- The mouse button you use to click with can also have an effect on what happens.
- The button to click will depend on your AUT and what the click should achieve.

Do not enter quotes around the mouse button numbers, en them in plain text.

Chapter 3

Overview of Components

The following diagram shows which abstract components are used and for which actions they can be used for.

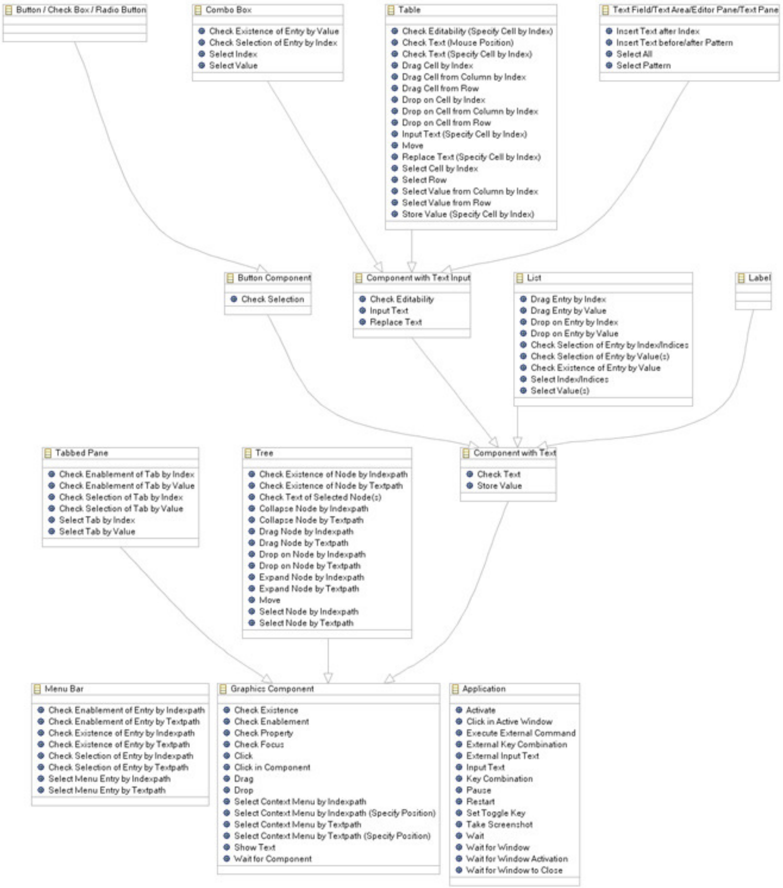

Button / Check Box / Radio Button
Combo Box
Check Existence of Entry by Value
Check Selection of Entry by Index
Select Index
Select Value
Table
Check Editability (Specify Cell by Index)
Check Text (Mouse Position)
Check Text (Specify Cell by Index)
Drag Cell by Index
Drag Cell from Column by Index
Drag Cell from Row
Drop on Cell by Index
Drop on Cell from Column by Index
Drop on Cell from Row
Input Text (Specify Cell by Index)
Move
Replace Text (Specify Cell by Index)
Select Cell by Index
Select Row
Select Value from Column by Index
Select Value from Row
Store Value (Specify Cell by Index)
Text Field/Text Area/Editor Pane/Text Pane
Insert Text after Index
Insert Text before/after Pattern
Select All
Select Pattern
Button Component
Check Selection
Component with Text Input
Check Editability
Input Text
Replace Text
List
Drag Entry by Index
Drag Entry by Value
Drop on Entry by Index
Drop on Entry by Value
Check Selection of Entry by Index/Indices
Check Selection of Entry by Value(s)
Check Existence of Entry by Value
Select Index/Indices
Select Value(s)
Label
Tabbed Pane
Check Enablement of Tab by Index
Check Enablement of Tab by Value
Check Selection of Tab by Index
Check Selection of Tab by Value
Select Tab by Index
Select Tab by Value
Tree
Check Existence of Node by Indexpath
Check Existence of Node by Textpath
Check Text of Selected Node(s)
Collapse Node by Indexpath
Collapse Node by Textpath
Drag Node by Indexpath
Drag Node by Textpath
Drop on Node by Indexpath
Drop on Node by Textpath
Expand Node by Indexpath
Expand Node by Textpath
Move
Select Node by Indexpath
Select Node by Textpath
Component with Text
Check Text
Store Value
Menu Bar
Check Enablement of Entry by Indexpath
Check Enablement of Entry by Textpath
Check Existence of Entry by Indexpath
Check Existence of Entry by Textpath
Check Selection of Entry by Indexpath
Check Selection of Entry by Textpath
Select Menu Entry by Indexpath
Select Menu Entry by Textpath
Graphics Component
Check Existence
Check Enablement
Check Property
Check Focus
Click
Click in Component
Drag
Drop
Select Context Menu by Indexpath
Select Context Menu by Indexpath (Specify Position)
Select Context Menu by Textpath
Select Context Menu by Textpath (Specify Position)
Show Text
Wait for Component
Application
Activate
Click in Active Window
Execute External Command
External Key Combination
External Input Text
Input Text
Key Combination
Pause
Restart
Set Toggle Key
Take Screenshot
Wait
Wait for Window
Wait for Window Activation
Wait for Window to Close

Figure 3.1: *Component structure*

Figure 3.1 provides an overview of the concrete and abstract component structure. This figure can be interpreted as follows: the arrows show inheritance relations between the components. Thus, a *Label* component has no actions of its own, but inherits them all from *Component with Text*. The *Label* component also inherits any actions that *Component with Text* also inherited. In this case, these are the actions from *Graphics component*.

Chapter 4

Index