

# Jubula Reference Manual

Jubula Team

BREDEX GmbH

June 29, 2012



BREDEX GmbH  
Mauernstr. 33  
38100 Braunschweig  
Germany  
Tel: +49-531 - 243 30 - 0  
Fax: +49-531 - 243 30 - 99  
[www.bredex.de](http://www.bredex.de)

**GUI***dancer* is a registered trademark of BREDEX GmbH

Title: Jubula Reference Manual  
Author: Jubula Team  
File: ReferenceManual  
State: RELEASE  
Version: V6.0.01011  
Released by: BREDEX GmbH  
Released at: June 29, 2012

# Contents

<b>1</b>	<b>Introduction</b>	<b>11</b>
<b>2</b>	<b>Shortcuts</b>	<b>13</b>
<b>3</b>	<b>Using Regular Expressions for Text Verification</b>	<b>15</b>
<b>4</b>	<b>Using Simple Match for Text Verification</b>	<b>19</b>
<b>5</b>	<b>Components, Actions, and Parameters</b>	<b>21</b>
5.1	abstract Toolkit . . . . .	21
5.1.1	Button Component . . . . .	22
5.1.1.1	Check Selection . . . . .	27
5.1.2	Component with Text . . . . .	28
5.1.2.1	Check Text . . . . .	32
5.1.2.2	Store Value . . . . .	33
5.1.3	Component with Text Input . . . . .	34
5.1.3.1	Check Editability . . . . .	39
5.1.3.2	Input Text . . . . .	39
5.1.3.3	Replace Text . . . . .	40
5.1.4	Graphics Component . . . . .	42
5.1.4.1	Check Enablement . . . . .	45
5.1.4.2	Check Enablement of Context Menu Entry by Indexpath . . .	46
5.1.4.3	Check Enablement of Context Menu Entry by Indexpath (Spec- ify Position) . . . . .	47
5.1.4.4	Check Enablement of Context Menu Entry by Textpath . . .	50
5.1.4.5	Check Enablement of Context Menu Entry by Textpath (Spec- ify Position) . . . . .	51
5.1.4.6	Check Existence . . . . .	54
5.1.4.7	Check Existence of Context Menu Entry by Indexpath . . . . .	55

5.1.4.8	Check Existence of Context Menu Entry by Indexpath (Specify Position) . . . . .	56
5.1.4.9	Check Existence of Context Menu Entry by Textpath . . . . .	59
5.1.4.10	Check Existence of Context Menu Entry by Textpath (Specify Position) . . . . .	60
5.1.4.11	Check Focus . . . . .	63
5.1.4.12	Check Property . . . . .	64
5.1.4.13	Check Selection of Context Menu Entry by Indexpath . . . . .	65
5.1.4.14	Check Selection of Context Menu Entry by Indexpath (Specify Position) . . . . .	67
5.1.4.15	Check Selection of Context Menu Entry by Textpath . . . . .	69
5.1.4.16	Check Selection of Context Menu Entry by Textpath (Specify Position) . . . . .	71
5.1.4.17	Click . . . . .	74
5.1.4.18	Click in Component . . . . .	75
5.1.4.19	Drag . . . . .	77
5.1.4.20	Drop . . . . .	80
5.1.4.21	Select Context Menu Entry by Indexpath . . . . .	82
5.1.4.22	Select Context Menu Entry by Indexpath (Specify Position) . . . . .	83
5.1.4.23	Select Context Menu Entry by Textpath . . . . .	86
5.1.4.24	Select Context Menu Entry by Textpath (Specify Position) . . . . .	88
5.1.4.25	Show Text . . . . .	91
5.1.4.26	Store Property . . . . .	92
5.1.4.27	Wait for Component . . . . .	93
5.2	concrete Toolkit . . . . .	93
5.2.1	Application . . . . .	94
5.2.1.1	Activate . . . . .	96
5.2.1.2	Check Existence of Window . . . . .	97
5.2.1.3	Check Numeric Values . . . . .	98
5.2.1.4	Check String Values . . . . .	99
5.2.1.5	Click in Active Window . . . . .	100
5.2.1.6	Copy Text to Clipboard . . . . .	102
5.2.1.7	Execute External Command . . . . .	103
5.2.1.8	External Input Text . . . . .	104

5.2.1.9	External Key Combination . . .	105
5.2.1.10	Input Text . . . . .	107
5.2.1.11	Key Combination . . . . .	108
5.2.1.12	Manual Test Step . . . . .	110
5.2.1.13	Pause Test Execution . . . . .	110
5.2.1.14	Read Timer . . . . .	111
5.2.1.15	Restart . . . . .	111
5.2.1.16	Set Toggle Key . . . . .	112
5.2.1.17	Start Timer . . . . .	113
5.2.1.18	Store Value . . . . .	114
5.2.1.19	Take Screenshot . . . . .	114
5.2.1.20	Take Screenshot of Active Win- dow . . . . .	116
5.2.1.21	Wait . . . . .	118
5.2.1.22	Wait for Window . . . . .	118
5.2.1.23	Wait for Window Activation .	120
5.2.1.24	Wait for Window to Close . .	121
5.2.2	Button/Check Box/Radio Button . . . .	123
5.2.3	Combo Box . . . . .	129
5.2.3.1	Check Existence of Entry by Value . . . . .	135
5.2.3.2	Check Selection of Entry by Index . . . . .	136
5.2.3.3	Select Entry by Index . . . . .	137
5.2.3.4	Select Entry by Value . . . . .	137
5.2.4	Label . . . . .	139
5.2.5	List . . . . .	145
5.2.5.1	Check Existence of Entry by Value . . . . .	151
5.2.5.2	Check Selection of Entry by Index/Indices . . . . .	152
5.2.5.3	Check Selection of Entry by Value(s) . . . . .	153
5.2.5.4	Drag Entry by Index . . . . .	154
5.2.5.5	Drag Entry by Value . . . . .	155
5.2.5.6	Drop on Entry by Index . . . .	158
5.2.5.7	Drop on Entry by Value . . . .	158
5.2.5.8	Select Entry by Index/Indices .	160
5.2.5.9	Select Entry by Value(s) . . . .	161
5.2.6	Menu Bar . . . . .	164
5.2.6.1	Check Enablement of Entry by Indexpath . . . . .	165
5.2.6.2	Check Enablement of Entry by Textpath . . . . .	166

5.2.6.3	Check Existence of Entry by Indexpath . . . . .	167
5.2.6.4	Check Existence of Entry by Textpath . . . . .	168
5.2.6.5	Check Selection of Entry by Indexpath . . . . .	169
5.2.6.6	Check Selection of Entry by Textpath . . . . .	169
5.2.6.7	Select Menu Entry by Indexpath	171
5.2.6.8	Select Menu Entry by Textpath	171
5.2.6.9	Wait for Component . . . . .	172
5.2.7	Tabbed Pane . . . . .	173
5.2.7.1	Check Enablement of Tab by Index . . . . .	179
5.2.7.2	Check Enablement of Tab by Value . . . . .	180
5.2.7.3	Check Existence of Tab . . . . .	181
5.2.7.4	Check Selection of Tab by Index	182
5.2.7.5	Check Selection of Tab by Value	182
5.2.7.6	Check Text of Tab by Index . . . . .	183
5.2.7.7	Select Tab by Index . . . . .	184
5.2.7.8	Select Tab by Value . . . . .	185
5.2.8	Table . . . . .	186
5.2.8.1	Check Editability (Specify Cell)	195
5.2.8.2	Check Editability of Cell (Mouse Position) . . . . .	197
5.2.8.3	Check Editability of Selected Cell . . . . .	197
5.2.8.4	Check Existence of Value in Column . . . . .	198
5.2.8.5	Check Existence of Value in Row . . . . .	200
5.2.8.6	Check Text (Mouse Position) . . . . .	201
5.2.8.7	Check Text (Specify Cell) . . . . .	202
5.2.8.8	Drag Cell . . . . .	204
5.2.8.9	Drag Cell from Column . . . . .	209
5.2.8.10	Drag Cell from Row . . . . .	212
5.2.8.11	Drop on Cell . . . . .	215
5.2.8.12	Drop on Cell from Column . . . . .	218
5.2.8.13	Drop on Cell from Row . . . . .	220
5.2.8.14	Input Text (Specify Cell) . . . . .	222
5.2.8.15	Move . . . . .	224
5.2.8.16	Replace Text (Specify Cell) . . . . .	227
5.2.8.17	Select Cell . . . . .	229
5.2.8.18	Select Value from Column . . . . .	233

5.2.8.19	Select Value from Row . . . .	236
5.2.8.20	Store Value (Specify Cell) . . .	238
5.2.8.21	Store Value at Mouse Position	240
5.2.9	Text Field/Text Area/Editor Pane/Text Pane	242
5.2.9.1	Insert Text after Index . . . . .	247
5.2.9.2	Insert Text before/after Pattern	248
5.2.9.3	Select All . . . . .	249
5.2.9.4	Select Pattern . . . . .	249
5.2.10	Tree . . . . .	251
5.2.10.1	Check Existence of Node by Indexpath . . . . .	258
5.2.10.2	Check Existence of Node by Textpath . . . . .	259
5.2.10.3	Check Text (Mouse Position) .	261
5.2.10.4	Check Text of Selected Node(s)	262
5.2.10.5	Collapse Node by Indexpath .	263
5.2.10.6	Collapse Node by Textpath . .	265
5.2.10.7	Drag Node by Indexpath . . .	267
5.2.10.8	Drag Node by Textpath . . . .	270
5.2.10.9	Drop on Node by Indexpath .	273
5.2.10.10	Drop on Node by Textpath . .	274
5.2.10.11	Expand Node by Indexpath . .	276
5.2.10.12	Expand Node by Textpath . .	278
5.2.10.13	Move . . . . .	280
5.2.10.14	Select Node by Indexpath . .	282
5.2.10.15	Select Node by Textpath . . .	284
5.2.10.16	Store Text at Mouse Position .	287
5.2.10.17	Store Text of Selected Node .	288
5.3	gef Toolkit . . . . .	288
5.3.1	Figure Canvas . . . . .	289
5.3.1.1	Check Figure Existence . . . .	297
5.3.1.2	Check Figure Property . . . .	298
5.3.1.3	Check Tool Existence . . . . .	300
5.3.1.4	Click Connection . . . . .	301
5.3.1.5	Click Figure . . . . .	304
5.3.1.6	Click in Figure . . . . .	305
5.3.1.7	Drag Figure . . . . .	308
5.3.1.8	Drop on Figure . . . . .	311
5.3.1.9	Select Tool . . . . .	314
5.4	html Toolkit . . . . .	315
5.4.1	Browser . . . . .	316
5.4.1.1	Go Back . . . . .	316
5.4.1.2	Open URL . . . . .	316
5.4.1.3	Reload page . . . . .	317
5.4.2	HTML Hyperlink . . . . .	318

5.4.2.1	Check URL . . . . .	323
5.5	rcp Toolkit . . . . .	324
5.6	swing Toolkit . . . . .	324
5.7	swt Toolkit . . . . .	324
5.7.1	(SWT) Tree . . . . .	325
5.7.1.1	Check Selection of Checkbox by Indxpath . . . . .	332
5.7.1.2	Check Selection of Checkbox by Textpath . . . . .	334
5.7.1.3	Check Selection of Checkbox on Selected Node . . . . .	336
5.7.1.4	Toggle Checkbox on Node by Indxpath . . . . .	336
5.7.1.5	Toggle Checkbox on Node by Textpath . . . . .	337
5.7.2	Toolbar Item . . . . .	340
5.7.2.1	Check Enablement of Entry by Indxpath . . . . .	346
5.7.2.2	Check Enablement of Entry by Textpath . . . . .	347
5.7.2.3	Check Existence of Entry by Indxpath . . . . .	348
5.7.2.4	Check Existence of Entry by Textpath . . . . .	349
5.7.2.5	Check Selection of Entry by Indxpath . . . . .	350
5.7.2.6	Check Selection of Entry by Textpath . . . . .	351
5.7.2.7	Select Menu Entry by Indxpath	352
5.7.2.8	Select Menu Entry by Textpath	353
5.7.3	Tree Table . . . . .	354
5.7.3.1	Check Selection by Value(s) (Spec- ify Column) . . . . .	363
5.7.3.2	Select Entry by Indxpath (Spec- ify Column) . . . . .	364
5.7.3.3	Select Entry by Textpath (Spec- ify Column) . . . . .	366
5.8	Deprecated Actions . . . . .	369
<b>6</b>	<b>Constants for lists, combo boxes, trees and menus</b>	<b>373</b>
6.1	Parameters which use Indices . . . . .	374
<b>7</b>	<b>Overview of Components</b>	<b>375</b>
<b>8</b>	<b>Using Relative Paths</b>	<b>377</b>

<b>9</b>	<b>Special characters in Jubula</b>	<b>379</b>
9.1	Verbatim text symbol . . . . .	379
9.2	General special characters . . . . .	379
9.3	Symbols with special meanings for certain pa- rameters . . . . .	381
<b>10</b>	<b>Language Codes</b>	<b>383</b>
<b>11</b>	<b>Keyboard layout files</b>	<b>387</b>
11.1	Creating a keyboard layout file . . . . .	387
<b>12</b>	<b>Debugging</b>	<b>389</b>
12.1	Remote Debugging . . . . .	389
12.1.1	Configuring Eclipse for remote debug- ging . . . . .	389
<b>13</b>	<b>Index</b>	<b>391</b>



# Chapter 1

## Introduction

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



# Chapter 2

## Shortcuts

Jubula offers various keyboard shortcuts to make working with the keyboard easier and quicker.

Most of the shortcuts can be changed in the *Preferences* in the ITE. Shortcuts for object mapping and observation are in the Jubula preferences. Other shortcuts can be seen and changed in the *General/Keys* preferences.

---

**Press »CTRL+SHIFT+L« twice to see the list of shortcuts used in Jubula and Eclipse.**

---





# Chapter 3

## Using Regular Expressions for Text Verification

- Using regular expressions lets you verify or select text even if you do not know exactly what the text is.
- Using a system of placeholders for characters and some function symbols, you can enter a regular expression to find or check a text.
- Actions which support regular expressions have an additional parameter, "*Operator*". From this combo box, you can choose "*matches*" to indicate that you want to use regular expressions.

### Simple matching

- `abc` matches "*abc*" and nothing else.

### Wildcards

- `.` represents one instance of any character.
- `..` represents two characters.

### Repetition

- Instead of using `.` for each character, you can use symbols to indicate how many characters you are replacing with wildcards.
- `?` matches the previous character or group 0 or 1 time(s).  
E.g. `a?` represents "none or one *a*".
- `*` matches the previous character or group 0 or more time(s).  
E.g. `a*` represents "none or more *a*".

- + matches the previous character or group 1 or more time(s).  
E.g. a+ represents "one or more a".
- You can also use curly brackets { } to specify the minimum and maximum number of repetitions, e.g. { 4 , 7 } looks for a minimum of four and a maximum of seven repetitions of the previous character.
- To specify that a character must be repeated a minimum of four times, use: { 4 , }

### **Combining wildcards with repetition**

- You can specify a whole area of unknown text using . and one of the repetition methods.
- . \* is an unlimited amount of any character, or none at all.
- . ? is 0 or one of any character.
- . + is an unlimited amount of any character, but the character must appear at least once.



---

**Jubula applies your regular expression to your entire string. To search for a match within a string, wildcards need to be placed on either side. See the examples below for more information.**

---

### **Ranges**

- For each individual character, you can specify a range of things it is allowed to be.
- A range is specified using square brackets ( [ ] ) and a dash -.
- For example, you can specify that a particular character can be any capital letter: [ A - Z ] .



---

**Note that there are no spaces between the ranges.**

---

### **Alternatives**

- Use a pipe ( ' | ' ) to specify alternatives.
- For example, [ a | b ] . \* will match a string that begins with a or b.

### Escape character

- Backslash `\` is used to negate the effect of the character following the backslash.
- The characters that are used to construct a regular expression need to be escaped if they are to be matched within a string.
- The characters are:  
`[ ] \ . | ? * + ( ) { } ^ $`

- Because Jubula already uses a backslash as an escape symbol, you will need to use two backslashes to escape regular expression characters.

- For example, to check for a tree node:

`x/y/z/**`

where the slashes are a part of the node, your regular expression in Jubula would look like this: `x\\/y\\/z\\/\\*\\*\\*`

The backslashes before the ordinary slashes are an escape symbol to tell Jubula that the following sign is not a path separator. The extra backslash before the stars tells Jubula that the second backslash is to be interpreted as a backslash in the regular expression, i.e. as an escape symbol.

- E.g. If you want to check for a star (\*), then you have to enter `\\*`.

### Verbatim

- You can avoid having to use multiple backslashes by putting the whole regular expression in single quotes:
- The example above for a tree node could be entered thus:  
`'x/y/z/**'`

### Useful examples

- An empty field is represented by: `^$`
- A string that starts with `a` is represented by: `a.*`
- A string that ends in `a` is represented by: `.*a`
- A string that starts with `a`, ends in `b` and has unknown values (0 or more) in the middle is represented by: `a.*b`

- A string which contains a somewhere between other unknown characters (0 or more) is represented by: `. * a . *`
- A password which can only contain capital letters and which must be between six and eight letters is represented by: `[ A - Z ] { 6 , 8 } .`
- A password which can contain any alphanumerical values and which must be at least six characters is represented by: `[ A - Z a - z 0 - 9 ] { 6 , } .`
- You can check that a string corresponds to a minimum and maximum length using:  
`' ^ . { ' = { MIN_COUNT } ' , ' = { MAX_COUNT } ' } $ '`
- To check for a text which begins with a `*`, you must use the escape character: `\\ * . *`
- To check that a specific string is *not* in a text, use:  
`^ . * [ ^ ( STRING ) ] . * \\ $`

For additional information about syntax and usage of regular expressions in general, please consult one of the many text-books on the subject.

## Chapter 4

# Using Simple Match for Text Verification

- Using *simple match* lets you verify or select text even if you do not know exactly what the text is.
- Actions which support regular expressions have an additional parameter, "Operator". From this combo box, you can choose "*simple match*" to indicate that you want to use simple matching.

---

***Simple match* is a good option if you are sure that you will only want to use the basic wildcards given below. For more powerful regular expressions, use the *matches* parameter ( → page 15) .**

---



### Simple matching

- `abc` matches "`abc`" and nothing else.

### Wildcards

- `?` represents one instance of any character.
- `*` represents any number (zero or more) of any characters.

### Useful examples

- A string that starts with `a` is represented by: `a*`
- A string that ends in `a` is represented by: `*a`
- A string that starts with `a`, ends in `b` and has unknown values (0 or more) in the middle is represented by: `a*b`

- A string which contains a somewhere between other unknown characters (0 or more) is represented by: `*a*`

# Chapter 5

## 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'.

### 5.1 abstract Toolkit

The abstract toolkit contains component types which are available on all toolkits supported by Jubula, 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 in Jubula to make Test Steps more general.

## 5.1.1 Button Component

### Description:

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

### Synopsis:

- Button Component (abstract)
  - Component with Text (abstract)
    - \* Graphics Component (abstract)

### New Actions

Name	Type	Parameters
Check Selection ( → page 27)	c	Boolean:Selected

### Inherited Actions

Name	Type	Parameters	Inherited from
Check Enablement ( → page 45)	c	Boolean:Enabled	Graphics Component (abstract)
Check Enablement of Context Menu Entry by Indxpath ( → page 46)	c	String:Indxpath Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)
Check Enablement of Context Menu Entry by Indxpath (Specify Position) ( → page 47)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Indxpath Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)

Check Enablement of Context Menu Entry by Textpath ( → page 50)	c	String:Textpath String:Operator Boolean:Enabled Integer:Mouse Button	Graphics component (abstract)	Component (abstract)
Check Enablement of Context Menu Entry by Textpath (Specify Position) ( → page 51)	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)	Component (abstract)
Check Existence ( → page 54)	c	Boolean:Exists	Graphics component (abstract)	Component (abstract)
Check Existence of Context Menu Entry by Indexpath ( → page 55)	c	String:Indexpath Boolean:Exists Integer:Mouse Button	Graphics component (abstract)	Component (abstract)
Check Existence of Context Menu Entry by Indexpath (Specify Position) ( → page 56)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Indexpath Boolean:Exists Integer:Mouse Button	Graphics component (abstract)	Component (abstract)
Check Existence of Context Menu Entry by Textpath ( → page 59)	c	String:Textpath String:Operator Boolean:Exists Integer:Mouse Button	Graphics component (abstract)	Component (abstract)
Check Existence of Context Menu Entry by Textpath (Specify Position) ( → page 60)	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)	Component (abstract)

Check Focus ( → page 63)	c	Boolean:Has Focus	Graphics Component (abstract)	Component (abstract)
Check Property ( → page 64)	c	String:PropertyName String:PropertyValue String:Operator	Graphics Component (abstract)	Component (abstract)
Check Selection of Context Menu Entry by Indexpath ( → page 65)	c	String:Indexpath Boolean:Selected Integer:MouseButton	Graphics Component (abstract)	Component (abstract)
Check Selection of Context Menu Entry by Indexpath (Specify Position) ( → page 67)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Indexpath Boolean:Selected Integer:MouseButton	Graphics Component (abstract)	Component (abstract)
Check Selection of Context Menu Entry by Textpath ( → page 69)	c	String:Textpath String:Operator Boolean:Selected Integer:MouseButton	Graphics Component (abstract)	Component (abstract)
Check Selection of Context Menu Entry by Textpath (Specify Position) ( → page 71)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Textpath String:Operator Boolean:Selected Integer:MouseButton	Graphics Component (abstract)	Component (abstract)

Check Text ( → page 32)	c	String:Text String:Operator	Component with Text (ab- stract)	
Click ( → page 74)	e	Integer:Number of Clicks Integer:Mouse Button	Graphics component (ab- stract)	Com- ponent (ab- stract)
Click in Component ( → page 75)	e	Integer:Number of Clicks Integer:Mouse Button Integer:x- position String:x-units Integer:y- position String:y-units	Graphics component (ab- stract)	Com- ponent (ab- stract)
Drag ( → page 77)	e	Integer:Mouse Button String:Modifier Keys Integer:x- position String:x-units Integer:y- position String:y-units	Graphics component (ab- stract)	Com- ponent (ab- stract)
Drop ( → page 80)	e	Integer:x- position String:x-units Integer:y- position String:y-units Integer:Delay before drop (milliseconds)	Graphics component (ab- stract)	Com- ponent (ab- stract)

Select Context Menu Entry by Indxpath ( → page 82)	e	String:Indxpath Integer:Mouse Button	Graphics Component (abstract)
Select Context Menu Entry by Indxpath (Specify Position) ( → page 83)	e	Integer:x-position String:x-units Integer:y-position String:y-units String:Indxpath Integer:Mouse Button	Graphics Component (abstract)
Select Context Menu Entry by Textpath ( → page 86)	e	String:Textpath String:Operator Integer:Mouse Button	Graphics Component (abstract)
Select Context Menu Entry by Textpath (Specify Position) ( → page 88)	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 91)	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 92)	e	Variable:Variable Name String:Property Name	Graphics Component (abstract)
Store Value ( → page 33)	e	Variable:Variable Name	Component with Text (abstract)

Wait for Component ( → page 93)	e	Integer:Timeout in ms Integer:Delay after Visibility	Graphics Component (abstract)
------------------------------------	---	---	----------------------------------

## 5.1.1.1 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

Button/Check Box/Radio Button (concrete)  
Toolbar Item (swt)

## 5.1.2 Component with Text

### Description:

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

### Synopsis:

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

### New Actions

Name	Type	Parameters
Check Text ( → page 32)	c	String:Text String:Operator
Store Value ( → page 33)	e	Variable:Variable Name

### Inherited Actions

Name	Type	Parameters	Inherited from
Check Enablement ( → page 45)	c	Boolean:Enabled	Graphics Component (abstract)
Check Enablement of Context Menu Entry by Indxpath ( → page 46)	c	String:Indxpath Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)
Check Enablement of Context Menu Entry by Indxpath (Specify Position) ( → page 47)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Indxpath Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)

Check Enablement of Context Menu Entry by Textpath ( → page 50)	c	String:Textpath String:Operator Boolean:Enabled Integer:Mouse Button	Graphics component (abstract)	Component (abstract)
Check Enablement of Context Menu Entry by Textpath (Specify Position) ( → page 51)	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)	Component (abstract)
Check Existence ( → page 54)	c	Boolean:Exists	Graphics component (abstract)	Component (abstract)
Check Existence of Context Menu Entry by Indexpath ( → page 55)	c	String:Indexpath Boolean:Exists Integer:Mouse Button	Graphics component (abstract)	Component (abstract)
Check Existence of Context Menu Entry by Indexpath (Specify Position) ( → page 56)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Indexpath Boolean:Exists Integer:Mouse Button	Graphics component (abstract)	Component (abstract)
Check Existence of Context Menu Entry by Textpath ( → page 59)	c	String:Textpath String:Operator Boolean:Exists Integer:Mouse Button	Graphics component (abstract)	Component (abstract)
Check Existence of Context Menu Entry by Textpath (Specify Position) ( → page 60)	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)	Component (abstract)

Check Focus ( → page 63)	c	Boolean:Has Focus	Graphics Component (abstract)	Component (abstract)
Check Property ( → page 64)	c	String:PropertyName String:PropertyValue String:Operator	Graphics Component (abstract)	Component (abstract)
Check Selection of Context Menu Entry by Indexpath ( → page 65)	c	String:Indexpath Boolean:Selected Integer:MouseButton	Graphics Component (abstract)	Component (abstract)
Check Selection of Context Menu Entry by Indexpath (Specify Position) ( → page 67)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Indexpath Boolean:Selected Integer:MouseButton	Graphics Component (abstract)	Component (abstract)
Check Selection of Context Menu Entry by Textpath ( → page 69)	c	String:Textpath String:Operator Boolean:Selected Integer:MouseButton	Graphics Component (abstract)	Component (abstract)
Check Selection of Context Menu Entry by Textpath (Specify Position) ( → page 71)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Textpath String:Operator Boolean:Selected Integer:MouseButton	Graphics Component (abstract)	Component (abstract)

Click ( → page 74)	e	Integer: Number of Clicks Integer: Mouse Button	Graphics Component (abstract)	Component (abstract)
Click in Component ( → page 75)	e	Integer: Number of Clicks Integer: Mouse Button Integer: x-position String: x-units Integer: y-position String: y-units	Graphics Component (abstract)	Component (abstract)
Drag ( → page 77)	e	Integer: Mouse Button String: Modifier Keys Integer: x-position String: x-units Integer: y-position String: y-units	Graphics Component (abstract)	Component (abstract)
Drop ( → page 80)	e	Integer: x-position String: x-units Integer: y-position String: y-units Integer: Delay before drop (milliseconds)	Graphics Component (abstract)	Component (abstract)
Select Context Menu Entry by Indexpath ( → page 82)	e	String: Indexpath Integer: Mouse Button	Graphics Component (abstract)	Component (abstract)
Select Context Menu Entry by Indexpath (Specify Position) ( → page 83)	e	Integer: x-position String: x-units Integer: y-position String: y-units String: Indexpath Integer: Mouse Button	Graphics Component (abstract)	Component (abstract)

Select Context Menu Entry by Textpath ( → page 86)	e	String:Textpath String:Operator Integer:Mouse Button	Graphics Component (abstract)
Select Context Menu Entry by Textpath (Specify Position) ( → page 88)	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 91)	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 92)	e	Variable:Variable Name String:Property Name	Graphics Component (abstract)
Wait for Component ( → page 93)	e	Integer:Timeout in ms Integer:Delay after Visibility	Graphics Component (abstract)

### 5.1.2.1 Check Text

(Component with Text)

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

#### 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.

## Components, Actions, and Parameters

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 15) .
- Select "*matches*" to use a regular expression ( → page 15) .
- Regardless of the operator used: If there is more than one match, the first found will be chosen.

### 5.1.2.2 Store Value

(Component with Text)

- 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 under-scores.
- 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

Button/Check Box/Radio Button (concrete)  
 Combo Box (concrete)  
 HTML Hyperlink (html)  
 Label (concrete)  
 List (concrete)  
 Table (concrete)  
 Text Field/Text Area/Editor Pane/Text Pane (concrete)  
 Toolbar Item (swt)

### 5.1.3 Component with Text Input

#### Description:

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

#### Synopsis:

- Component with Text Input (abstract)
  - Component with Text (abstract)
    - \* Graphics Component (abstract)

#### New Actions

Name	Type	Parameters
Check Editability ( → page 39)	c	Boolean:Editable
Input Text ( → page 39)	e	String:Text
Replace Text ( → page 40)	e	String:Text

#### Inherited Actions

Name	Type	Parameters	Inherited from
Check Enablement ( → page 45)	c	Boolean:Enabled	Graphics Component (abstract)
Check Enablement of Context Menu Entry by Indxpath ( → page 46)	c	String:Indxpath Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)
Check Enablement of Context Menu Entry by Indxpath (Specify Position) ( → page 47)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Indxpath Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)

Check Enablement of Context Menu Entry by Textpath ( → page 50)	c	String:Textpath String:Operator Boolean:Enabled Integer:Mouse Button	Graphics component (abstract)	Component (abstract)
Check Enablement of Context Menu Entry by Textpath (Specify Position) ( → page 51)	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)	Component (abstract)
Check Existence ( → page 54)	c	Boolean:Exists	Graphics component (abstract)	Component (abstract)
Check Existence of Context Menu Entry by Indexpath ( → page 55)	c	String:Indexpath Boolean:Exists Integer:Mouse Button	Graphics component (abstract)	Component (abstract)
Check Existence of Context Menu Entry by Indexpath (Specify Position) ( → page 56)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Indexpath Boolean:Exists Integer:Mouse Button	Graphics component (abstract)	Component (abstract)
Check Existence of Context Menu Entry by Textpath ( → page 59)	c	String:Textpath String:Operator Boolean:Exists Integer:Mouse Button	Graphics component (abstract)	Component (abstract)
Check Existence of Context Menu Entry by Textpath (Specify Position) ( → page 60)	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)	Component (abstract)

Check Focus ( → page 63)	c	Boolean:Has Focus	Graphics Component (abstract)	Component (abstract)
Check Property ( → page 64)	c	String:PropertyName String:PropertyValue String:Operator	Graphics Component (abstract)	Component (abstract)
Check Selection of Context Menu Entry by Indexpath ( → page 65)	c	String:Indexpath Boolean:Selected Integer:MouseButton	Graphics Component (abstract)	Component (abstract)
Check Selection of Context Menu Entry by Indexpath (Specify Position) ( → page 67)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Indexpath Boolean:Selected Integer:MouseButton	Graphics Component (abstract)	Component (abstract)
Check Selection of Context Menu Entry by Textpath ( → page 69)	c	String:Textpath String:Operator Boolean:Selected Integer:MouseButton	Graphics Component (abstract)	Component (abstract)
Check Selection of Context Menu Entry by Textpath (Specify Position) ( → page 71)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Textpath String:Operator Boolean:Selected Integer:MouseButton	Graphics Component (abstract)	Component (abstract)

Check Text ( → page 32)	c	String:Text String:Operator	Component with Text (ab- stract)	
Click ( → page 74)	e	Integer:Number of Clicks Integer:Mouse Button	Graphics Component (ab- stract)	Com- ponent (ab- stract)
Click in Component ( → page 75)	e	Integer:Number of Clicks Integer:Mouse Button Integer:x- position String:x-units Integer:y- position String:y-units	Graphics Component (ab- stract)	Com- ponent (ab- stract)
Drag ( → page 77)	e	Integer:Mouse Button String:Modifier Keys Integer:x- position String:x-units Integer:y- position String:y-units	Graphics Component (ab- stract)	Com- ponent (ab- stract)
Drop ( → page 80)	e	Integer:x- position String:x-units Integer:y- position String:y-units Integer:Delay before drop (milliseconds)	Graphics Component (ab- stract)	Com- ponent (ab- stract)

Select Context Menu Entry by Indexpath ( → page 82)	e	String:Indexpath Integer:Mouse Button	Graphics Component (abstract)
Select Context Menu Entry by Indexpath (Specify Position) ( → page 83)	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 86)	e	String:Textpath String:Operator Integer:Mouse Button	Graphics Component (abstract)
Select Context Menu Entry by Textpath (Specify Position) ( → page 88)	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 91)	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 92)	e	Variable:Variable Name String:Property Name	Graphics Component (abstract)
Store Value ( → page 33)	e	Variable:Variable Name	Component with Text (abstract)

Wait for Component ( → page 93)	e	Integer:Timeout in ms Integer:Delay after Visibility	Graphics Component (abstract)	Component (abstract)
------------------------------------	---	---	----------------------------------	-------------------------

## 5.1.3.1 Check Editability

(Component with Text Input)

- 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.

## 5.1.3.2 Input Text

(Component with Text Input)

- Use this action to enter text into a component.
- The text you want to enter is given as a parameter.
- Jubula realizes this 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.

### 5.1.3.3 Replace Text

(Component with Text Input)

- Use this action to replace any text already in the text field.
- The text you want to enter is given as a parameter.
- Jubula realizes this 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**

#### 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 Box (concrete)

Table (concrete)

Text Field/Text Area/Editor Pane/Text Pane (concrete)

## 5.1.4 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 45)	c	Boolean:Enabled
Check Enablement of Context Menu Entry by Indxpath ( → page 46)	c	String:Indxpath Boolean:Enabled Integer:Mouse Button
Check Enablement of Context Menu Entry by Indxpath (Specify Posi- tion) ( → page 47)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Indxpath Boolean:Enabled Integer:Mouse Button
Check Enablement of Context Menu Entry by Textpath ( → page 50)	c	String:Textpath String:Operator Boolean:Enabled Integer:Mouse Button
Check Enablement of Context Menu Entry by Textpath (Specify Posi- tion) ( → page 51)	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 54)	c	Boolean:Exists
Check Existence of Context Menu Entry by Indexpath ( → page 55)	c	String:Indexpath Boolean:Exists Integer:Mouse Button
Check Existence of Context Menu Entry by Indexpath (Specify Position) ( → page 56)	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 59)	c	String:Textpath String:Operator Boolean:Exists Integer:Mouse Button
Check Existence of Context Menu Entry by Textpath (Specify Position) ( → page 60)	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 63)	c	Boolean:Has Focus
Check Property ( → page 64)	c	String:Property Name String:Property Value String:Operator
Check Selection of Context Menu Entry by Indexpath ( → page 65)	c	String:Indexpath Boolean:Selected Integer:Mouse Button
Check Selection of Context Menu Entry by Indexpath (Specify Position) ( → page 67)	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 69)	c	String:Textpath String:Operator Boolean:Selected Integer:Mouse Button
Check Selection of Context Menu Entry by Textpath (Specify Position) ( → page 71)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Textpath String:Operator Boolean:Selected Integer:Mouse Button
Click ( → page 74)	e	Integer:Number of Clicks Integer:Mouse Button
Click in Component ( → page 75)	e	Integer:Number of Clicks Integer:Mouse Button Integer:x-position String:x-units Integer:y-position String:y-units
Drag ( → page 77)	e	Integer:Mouse Button String:Modifier Keys Integer:x-position String:x-units Integer:y-position String:y-units
Drop ( → page 80)	e	Integer:x-position String:x-units Integer:y-position String:y-units Integer:Delay before drop (milliseconds)
Select Context Menu Entry by Indxpath ( → page 82)	e	String:Indxpath Integer:Mouse Button
Select Context Menu Entry by Indxpath (Specify Position) ( → page 83)	e	Integer:x-position String:x-units Integer:y-position String:y-units String:Indxpath Integer:Mouse Button

Select Context Menu Entry by Textpath ( → page 86)	e	String:Textpath String:Operator Integer:Mouse Button
Select Context Menu Entry by Textpath (Specify Position) ( → page 88)	e	Integer:x-position String:x-units Integer:y-position String:y-units String:Menupath String:Operator Integer:Mouse Button
Show Text ( → page 91)	e	String:Text Integer:Text Size (in points) Integer:Time per Word (in milliseconds) Integer:Window Width (in pixels)
Store Property ( → page 92)	e	Variable:Variable Name String:Property Name
Wait for Component ( → page 93)	e	Integer:Timeout in ms Integer:Delay after Visibility

## Inherited Actions

*none*

### 5.1.4.1 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.

### 5.1.4.2 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, enter them in plain text.**

---



### 5.1.4.3 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 tell Jubula 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 tell Jubula 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, enter them in plain text.**

---



#### 5.1.4.4 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 15) .
- Select "*matches*" to use a regular expression ( → page 15) .
- Regardless of the operator used: If there is more than one match, the first found will be chosen.

Name	Data Type	Values	Default
Enabled	Boolean	true false	true

## Components, Actions, and Parameters

- 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, enter them in plain text.**

---



### 5.1.4.5 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 tell Jubula 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 tell Jubula 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 15) .

- Select "*matches*" to use a regular expression ( → page 15) .
- Regardless of the operator used: If there is more than one match, the first 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.**

#### 5.1.4.6 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.

### 5.1.4.7 Check Existence of Context Menu Entry by Indxpath

(Graphics Component)

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

---

**This action is unsupported for the HTML toolkit**

---



#### Parameters

Name	Data Type	Values	Default
Indxpath	String	–	none

- Enter the path to the item as an indxpath.
- 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.**

---

#### 5.1.4.8 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 tell Jubula 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 tell Jubula 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.**

---

## 5.1.4.9 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 15) .
- Select "*matches*" to use a regular expression ( → page 15) .
- Regardless of the operator used: If there is more than one match, the first 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.**

#### 5.1.4.10 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 tell Jubula 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 tell Jubula 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 15) .

- Select "*matches*" to use a regular expression ( → page 15) .
- Regardless of the operator used: If there is more than one match, the first 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.**



### 5.1.4.11 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.

## 5.1.4.12 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.



**If the property is *private*, it cannot be checked, as Jubula 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

## Components, Actions, and Parameters

- 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 15) .
- Select "*matches*" to use a regular expression ( → page 15) .
- Regardless of the operator used: If there is more than one match, the first found will be chosen.

### 5.1.4.13 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.**

---

### 5.1.4.14 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 tell Jubula 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 tell Jubula 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, enter them in plain text.**

---



### 5.1.4.15 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 15) .
- Select "*matches*" to use a regular expression ( → page 15) .
- Regardless of the operator used: If there is more than one match, the first 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.

## Components, Actions, and Parameters

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.**

---



### 5.1.4.16 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 tell Jubula 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 tell Jubula 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 15) .
- Select "*matches*" to use a regular expression ( → page 15) .
- Regardless of the operator used: If there is more than one match, the first 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.**

#### 5.1.4.17 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.

## Components, Actions, and 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.**



### 5.1.4.18 Click in Component

(Graphics Component)

- This item sends one or more mouse clicks to the component.
- Jubula realises this 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 tell Jubula 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 tell Jubula 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.

### 5.1.4.19 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**

## Components, Actions, and 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 tell Jubula 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 tell Jubula 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.

#### 5.1.4.20 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.**

## Components, Actions, and Parameters

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

- Use this parameter to tell Jubula 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 tell Jubula 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.

### 5.1.4.21 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.
- Jubula realises this 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, enter them in plain text.**

---



### 5.1.4.22 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.
- Jubula realises this 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 tell Jubula 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 tell Jubula 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.**

---

#### 5.1.4.23 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.
- Jubula realises this 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 15) .
- Select "*matches*" to use a regular expression ( → page 15) .
- Regardless of the operator used: If there is more than one match, the first 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, enter them in plain text.**



#### 5.1.4.24 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.
- Jubula realises this 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 tell Jubula 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 tell Jubula 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 15) .
- Select "*matches*" to use a regular expression ( → page 15) .
- Regardless of the operator used: If there is more than one match, the first 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, enter them in plain text.**



### 5.1.4.25 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.

### 5.1.4.26 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.



**If the property is *private*, it cannot be stored, as Jubula 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 under-scores.
- 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.

## 5.1.4.27 Wait for Component

(Graphics Component)

- The wait for component action lets you choose a component which you need for the next steps.
- Jubula 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) Jubula should 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

(SWT) Tree (swt)  
 Button/Check Box/Radio Button (concrete)  
 Combo Box (concrete)  
 Figure Canvas (gef)  
 HTML Hyperlink (html)  
 Label (concrete)  
 List (concrete)  
 Tabbed Pane (concrete)  
 Table (concrete)  
 Text Field/Text Area/Editor Pane/Text Pane (concrete)  
 Toolbar Item (swt)  
 Tree (concrete)  
 Tree Table (swt)

## 5.2 concrete Toolkit

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

## 5.2.1 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 96)	e	String:Activation Method
Check Existence of Window ( → page 97)	e	String:Title String:Operator Boolean:Exists
Check Numeric Values ( → page 98)	e	String:Value 1 String:Comparison Method String:Value 2
Check String Values ( → page 99)	e	String:Value 1 String:Value 2 Operator String:Value 2
Click in Active Window ( → page 100)	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 102)	e	String:Text
Execute External Command ( → page 103)	e	String:Command Integer:Expected Exit Code Boolean:Local Integer:Timeout in ms
External Input Text ( → page 104)	e	String:Text
External Key Combination ( → page 105)	e	String:Modifier Keys String:Base Key

Input Text ( → page 107)	e	String:Text
Key Combination ( → page 108)	e	String:Modifier Keys String:Base Key
Manual Test Step ( → page 110)	e	String:Action to perform String:Expected Behavior Integer:Timeout in ms
Pause Test Execution ( → page 110)	e	
Read Timer ( → page 111)	e	String:Timer Name Variable:Timer Value Variable Name
Restart ( → page 111)	e	
Set Toggle Key ( → page 112)	e	Integer:Key Boolean:Activate Key
Start Timer ( → page 113)	e	String:Timer Name Variable:Start Time Variable Name
Store Value ( → page 114)	e	Variable:Variable Name String:Value
Take Screenshot ( → page 114)	e	String:Destination Integer:Delay String:File Access Integer:Scaling Factor Boolean:Create Directories
Take Screenshot of Active Window ( → page 116)	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 118)	e	Integer:Millisecs
Wait for Window ( → page 118)	e	String:Title String:Operator Integer:Timeout in ms Integer:Delay after Visibility
Wait for Window Acti- vation ( → page 120)	e	String:Title String:Operator Integer:Timeout in ms Integer:Delay after Visibility
Wait for Window to Close ( → page 121)	e	String:Title String:Operator Integer:Timeout in ms Integer:Delay after Closure

### Inherited Actions

*none*

#### 5.2.1.1 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.
- Jubula realises this action by clicking in the location indicated in the *activation method* parameter.
- In the AUT configuration (→ *User Manual* p. 48), 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.

## 5.2.1.2 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 15) .
- Select "*matches*" to use a regular expression ( → page 15) .
- Regardless of the operator used: If there is more than one match, the first 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.

### 5.2.1.3 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.

## 5.2.1.4 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 15) .
- Select "*matches*" to use a regular expression ( → page 15) .
- Regardless of the operator used: If there is more than one match, the first 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*.

### 5.2.1.5 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

## Components, Actions, and Parameters

- 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) positions.**

---



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

- Use this parameter to tell Jubula 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.**

---



---

**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 tell Jubula 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.

### 5.2.1.6 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 379) .

## 5.2.1.7 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 that Jubula should wait for, as well as a timeout to tell Jubula how long to wait for the code.

### Parameters

Name	Data Type	Values	Default
Command	String	–	none

- Use this parameter to tell Jubula 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\guidancer\guidancer.exe  
-data C:\Program Files\guidancer\ws  
enter:  
"C:\Program Files\guidancer\guidancer.exe"  
-data "C:\Program Files\guidancer\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 successful).
- Use this parameter to define the exit code that Jubula should expect.

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, the same machine as Jubula is installed on) 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, there is no difference between running the command locally or remotely.**

---

- For information on using relative paths to the location of the commands on local and remote machines, see the section in this document ( → page 377) .

Name	Data Type	Values	Default
Timeout in ms	Integer	–	5000

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

### 5.2.1.8 External Input Text

(Application)

- Use this action to input text into dialogs that are not supported by Jubula (e.g. file choosers).
- Jubula can send keystrokes to these dialogs, but cannot check 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.

### 5.2.1.9 External Key Combination

(Application)

- Use this action to send a key combination to a dialog that is not supported by Jubula (e.g. file choosers).
- Jubula can send a keystroke to these dialogs, but cannot check if it have 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 text**

Name	Data Type	Values	Default
Base Key	String	–	none

## Components, Actions, and Parameters

- 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 <code>shift</code> 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.

### 5.2.1.10 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 Jubula can check that the keystrokes have arrived at the component.
- Use this action for components that Jubula supports, 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.

## 5.2.1.11 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 <code>shift</code> 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.

### 5.2.1.12 Manual Test Step

(Application)

- Use this action to specify a manual Test Step.
- Manual Test Steps can be specified in Jubula and must be executed by a manual tester.
- Jubula provides 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 Steps than for automated ones, as a human tester may need more time to perform the action(s) and document any errors.**

### 5.2.1.13 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.

## 5.2.1.14 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.

## 5.2.1.15 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.**

---

This action has no parameters.

#### 5.2.1.16 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, Jubula does not support this action 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 tell Jubula 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.

### 5.2.1.17 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.

### 5.2.1.18 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 under-scores.
- 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.

### 5.2.1.19 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

## Components, Actions, and Parameters

- 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 377) .

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.

### 5.2.1.20 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 the 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 377) .

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.

#### 5.2.1.21 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.
- Jubula also offers 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.

#### 5.2.1.22 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 15) .
- Select "*matches*" to use a regular expression ( → page 15) .
- Regardless of the operator used: If there is more than one match, the first 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.

### 5.2.1.23 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 15) .
- Select "*matches*" to use a regular expression ( → page 15) .
- Regardless of the operator used: If there is more than one match, the first 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.

## 5.2.1.24 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 15) .
- Select "*matches*" to use a regular expression ( → page 15) .
- Regardless of the operator used: If there is more than one match, the first 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.

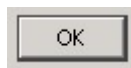
Name	Data Type	Values	Default
Delay after Closure	Integer	–	200

- 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.

## 5.2.2 Button/Check Box/Radio Button

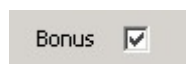
### Description:

- 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 5.1:** Button

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



**Figure 5.2:** Checkbox

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



**Figure 5.3:** Radio Button

### Synopsis:

- Button/Check Box/Radio Button (concrete)
  - Button Component (abstract)
    - \* Component with Text (abstract)
      - Graphics Component (abstract)

### New Actions

none

### Inherited Actions

Name	Type	Parameters	Inherited from
Check Enablement ( → page 45)	c	Boolean:Enabled	Graphics Component (abstract)

Check Enablement of Context Menu Entry by Indxpath ( → page 46)	c	String:Indxpath Boolean:Enabled Integer:Mouse Button	Graphics component (abstract)	Component (abstract)
Check Enablement of Context Menu Entry by Indxpath (Specify Position) ( → page 47)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Indxpath Boolean:Enabled Integer:Mouse Button	Graphics component (abstract)	Component (abstract)
Check Enablement of Context Menu Entry by Textpath ( → page 50)	c	String:Textpath String:Operator Boolean:Enabled Integer:Mouse Button	Graphics component (abstract)	Component (abstract)
Check Enablement of Context Menu Entry by Textpath (Specify Position) ( → page 51)	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)	Component (abstract)
Check Existence ( → page 54)	c	Boolean:Exists	Graphics component (abstract)	Component (abstract)
Check Existence of Context Menu Entry by Indxpath ( → page 55)	c	String:Indxpath Boolean:Exists Integer:Mouse Button	Graphics component (abstract)	Component (abstract)
Check Existence of Context Menu Entry by Indxpath (Specify Position) ( → page 56)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Indxpath Boolean:Exists Integer:Mouse Button	Graphics component (abstract)	Component (abstract)

Check Existence of Context Menu Entry by Textpath ( → page 59)	c	String:Textpath String:Operator Boolean:Exists Integer:Mouse Button	Graphics Component (abstract)	
Check Existence of Context Menu Entry by Textpath (Specify Position) ( → page 60)	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 63)	c	Boolean:Has Focus	Graphics Component (abstract)	
Check Property ( → page 64)	c	String:Property Name String:Property Value String:Operator	Graphics Component (abstract)	
Check Selection ( → page 27)	c	Boolean:Selected	Button Component (abstract)	
Check Selection of Context Menu Entry by Indexpath ( → page 65)	c	String:Indexpath Boolean:Selected Integer:Mouse Button	Graphics Component (abstract)	
Check Selection of Context Menu Entry by Indexpath (Specify Position) ( → page 67)	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 69)	c	String:Textpath String:Operator Boolean:Selected Integer:Mouse Button	Graphics Component (abstract)
Check Selection of Context Menu Entry by Textpath (Specify Position) ( → page 71)	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 32)	c	String:Text String:Operator	Component with Text (abstract)
Click ( → page 74)	e	Integer:Number of Clicks Integer:Mouse Button	Graphics Component (abstract)
Click in Component ( → page 75)	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 77)	e	Integer:Mouse Button String:Modifier Keys Integer:x-position String:x-units Integer:y-position String:y-units	Graphics Component (abstract)

Drop ( → page 80)	e	Integer:x-position String:x-units Integer:y-position String:y-units Integer:Delay before drop (milliseconds)	Graphics component (abstract)	Component (abstract)
Select Context Menu Entry by Indexpath ( → page 82)	e	String:Indexpath Integer:Mouse Button	Graphics component (abstract)	Component (abstract)
Select Context Menu Entry by Indexpath (Specify Position) ( → page 83)	e	Integer:x-position String:x-units Integer:y-position String:y-units String:Indexpath Integer:Mouse Button	Graphics component (abstract)	Component (abstract)
Select Context Menu Entry by Textpath ( → page 86)	e	String:Textpath String:Operator Integer:Mouse Button	Graphics component (abstract)	Component (abstract)
Select Context Menu Entry by Textpath (Specify Position) ( → page 88)	e	Integer:x-position String:x-units Integer:y-position String:y-units String:Menupath String:Operator Integer:Mouse Button	Graphics component (abstract)	Component (abstract)

Show Text ( → page 91)	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 92)	e	Variable:Variable Name String:Property Name	Graphics Component (abstract)
Store Value ( → page 33)	e	Variable:Variable Name	Component with Text (abstract)
Wait for Component ( → page 93)	e	Integer:Timeout in ms Integer:Delay after Visibility	Graphics Component (abstract)

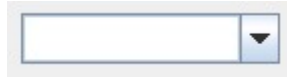
### Used By

Toolbar Item (swt)

### 5.2.3 Combo Box

#### Description:

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



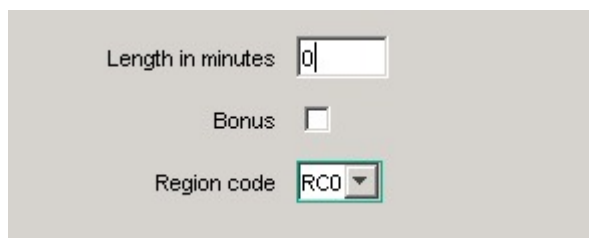
**Figure 5.4:** Combo Box

- 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 379) for more details.

#### Mapping combo boxes

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



**Figure 5.5:** 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 Box (concrete)
  - Component with Text Input (abstract)
    - \* Component with Text (abstract)
      - Graphics Component (abstract)

### New Actions

Name	Type	Parameters
Check Existence of Entry by Value ( → page 135)	c	String:Text String:Operator Boolean:Exists
Check Selection of Entry by Index ( → page 136)	c	String:Index Boolean:Selected
Select Entry by Index ( → page 137)	e	String:Index
Select Entry by Value ( → page 137)	e	String:Text String:Operator String:Search Type

## Inherited Actions

Name	Type	Parameters	Inherited from
Check Editability ( → page 39)	c	Boolean:Editable	Component with Text Input (abstract)
Check Enablement ( → page 45)	c	Boolean:Enabled	Graphics Component (abstract)
Check Enablement of Context Menu Entry by Indxpath ( → page 46)	c	String:Indxpath Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)
Check Enablement of Context Menu Entry by Indxpath (Specify Position) ( → page 47)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Indxpath Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)
Check Enablement of Context Menu Entry by Textpath ( → page 50)	c	String:Textpath String:Operator Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)
Check Enablement of Context Menu Entry by Textpath (Specify Position) ( → page 51)	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 54)	c	Boolean:Exists	Graphics Component (abstract)
Check Existence of Context Menu Entry by Indxpath ( → page 55)	c	String:Indxpath Boolean:Exists Integer:Mouse Button	Graphics Component (abstract)

Check Existence of Context Menu Entry by Indexpath (Specify Position) ( → page 56)	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 59)	c	String:Textpath String:Operator Boolean:Exists Integer:Mouse Button	Graphics Component (abstract)	
Check Existence of Context Menu Entry by Textpath (Specify Position) ( → page 60)	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 63)	c	Boolean:Has Focus	Graphics Component (abstract)	
Check Property ( → page 64)	c	String:Property Name String:Property Value String:Operator	Graphics Component (abstract)	
Check Selection of Context Menu Entry by Indexpath ( → page 65)	c	String:Indexpath Boolean:Selected Integer:Mouse Button	Graphics Component (abstract)	
Check Selection of Context Menu Entry by Indexpath (Specify Position) ( → page 67)	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 69)	c	String:Textpath String:Operator Boolean:Selected Integer:Mouse Button	Graphics Component (abstract)	
Check Selection of Context Menu Entry by Textpath (Specify Position) ( → page 71)	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 32)	c	String:Text String:Operator	Component with Text (abstract)	
Click ( → page 74)	e	Integer:Number of Clicks Integer:Mouse Button	Graphics Component (abstract)	
Click in Component ( → page 75)	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 77)	e	Integer:Mouse Button String:Modifier Keys Integer:x-position String:x-units Integer:y-position String:y-units	Graphics Component (abstract)	

Drop ( → page 80)	e	Integer:x-position String:x-units Integer:y-position String:y-units Integer:Delay before drop (milliseconds)	Graphics Component (abstract)
Input Text ( → page 39)	e	String:Text	Component with Text Input (abstract)
Replace Text ( → page 40)	e	String:Text	Component with Text Input (abstract)
Select Context Menu Entry by Indexpath ( → page 82)	e	String:Indexpath Integer:Mouse Button	Graphics Component (abstract)
Select Context Menu Entry by Indexpath (Specify Position) ( → page 83)	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 86)	e	String:Textpath String:Operator Integer:Mouse Button	Graphics Component (abstract)
Select Context Menu Entry by Textpath (Specify Position) ( → page 88)	e	Integer:x-position String:x-units Integer:y-position String:y-units String:Menuopath String:Operator Integer:Mouse Button	Graphics Component (abstract)

Show Text ( → page 91)	e	String:Text Integer:Text Size (in points) Integer:Time per Word (in milliseconds) Integer:Window Width (in pixels)	Graphics Component (ab- stract)	Com- ponent (ab- stract)
Store Property ( → page 92)	e	Variable:Variable Name String:Property Name	Graphics Component (ab- stract)	Com- ponent (ab- stract)
Store Value ( → page 33)	e	Variable:Variable Name	Component with Text (ab- stract)	(ab- stract)
Wait for Component ( → page 93)	e	Integer:Timeout in ms Integer:Delay after Visibility	Graphics Component (ab- stract)	Com- ponent (ab- stract)

## 5.2.3.1 Check Existence of Entry by Value

(Combo Box)

- 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 15) .
- Select "*matches*" to use a regular expression ( → page 15) .
- Regardless of the operator used: If there is more than one match, the first 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.

### 5.2.3.2 Check Selection of Entry by Index

(Combo Box)

- 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.

### 5.2.3.3 Select Entry by Index

(Combo Box)

- Use this action to select an item from the combo box using its index as a parameter.
- Jubula realises this 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.

### 5.2.3.4 Select Entry by Value

(Combo Box)

- Use this action to select an item from the combo box using the name of the item as a parameter.
- Jubula realises this 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 15) .
- Select "*matches*" to use a regular expression ( → page 15) .
- Regardless of the operator used: If there is more than one match, the first 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 occurrence of a value, ignoring all elements before the selected entry.

## 5.2.4 Label

### Description:

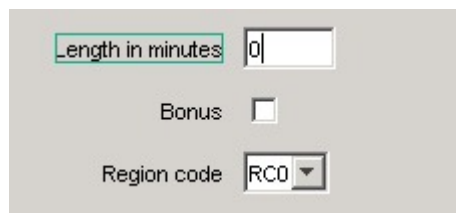
A *label* is used for displaying a small, descriptive text and/or an image.



**Figure 5.6:** Label

### Mapping labels

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



**Figure 5.7:** Label

### Synopsis:

- Label (concrete)
  - Component with Text (abstract)
  - \* Graphics Component (abstract)

### New Actions

none

### Inherited Actions

Name	Type	Parameters	Inherited from
Check Enablement ( → page 45)	c	Boolean:Enabled	Graphics Component (abstract)
Check Enablement of Context Menu Entry by Indxpath ( → page 46)	c	String:Indxpath Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)

Check Enablement of Context Menu Entry by Indexpath (Specify Position) ( → page 47)	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 50)	c	String:Textpath String:Operator Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)	
Check Enablement of Context Menu Entry by Textpath (Specify Position) ( → page 51)	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 54)	c	Boolean:Exists	Graphics Component (abstract)	
Check Existence of Context Menu Entry by Indexpath ( → page 55)	c	String:Indexpath Boolean:Exists Integer:Mouse Button	Graphics Component (abstract)	
Check Existence of Context Menu Entry by Indexpath (Specify Position) ( → page 56)	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 59)	c	String:Textpath String:Operator Boolean:Exists Integer:Mouse Button	Graphics Component (abstract)	
Check Existence of Context Menu Entry by Textpath (Specify Position) ( → page 60)	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 63)	c	Boolean:Has Focus	Graphics Component (abstract)	
Check Property ( → page 64)	c	String:Property Name String:Property Value String:Operator	Graphics Component (abstract)	
Check Selection of Context Menu Entry by Indexpath ( → page 65)	c	String:Indexpath Boolean:Selected Integer:Mouse Button	Graphics Component (abstract)	
Check Selection of Context Menu Entry by Indexpath (Specify Position) ( → page 67)	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 69)	c	String:Textpath String:Operator Boolean:Selected Integer:Mouse Button	Graphics Component (abstract)
Check Selection of Context Menu Entry by Textpath (Specify Position) ( → page 71)	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 32)	c	String:Text String:Operator	Component with Text (abstract)
Click ( → page 74)	e	Integer:Number of Clicks Integer:Mouse Button	Graphics Component (abstract)
Click in Component ( → page 75)	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 77)	e	Integer:Mouse Button String:Modifier Keys Integer:x-position String:x-units Integer:y-position String:y-units	Graphics Component (abstract)

Drop ( → page 80)	e	Integer:x-position String:x-units Integer:y-position String:y-units Integer:Delay before drop (milliseconds)	Graphics component (abstract)	Component (abstract)
Select Context Menu Entry by Indexpath ( → page 82)	e	String:Indexpath Integer:Mouse Button	Graphics component (abstract)	Component (abstract)
Select Context Menu Entry by Indexpath (Specify Position) ( → page 83)	e	Integer:x-position String:x-units Integer:y-position String:y-units String:Indexpath Integer:Mouse Button	Graphics component (abstract)	Component (abstract)
Select Context Menu Entry by Textpath ( → page 86)	e	String:Textpath String:Operator Integer:Mouse Button	Graphics component (abstract)	Component (abstract)
Select Context Menu Entry by Textpath (Specify Position) ( → page 88)	e	Integer:x-position String:x-units Integer:y-position String:y-units String:Menupath String:Operator Integer:Mouse Button	Graphics component (abstract)	Component (abstract)

Show Text ( → page 91)	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 92)	e	Variable:Variable Name String:Property Name	Graphics Component (abstract)
Store Value ( → page 33)	e	Variable:Variable Name	Component with Text (abstract)
Wait for Component ( → page 93)	e	Integer:Timeout in ms Integer:Delay after Visibility	Graphics Component (abstract)

## 5.2.5 List

### 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 379) for more details.



**Figure 5.8:** *List*

### Mapping lists

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



**Figure 5.9:** *List*

### Synopsis:

- List (concrete)
  - Component with Text (abstract)
    - \* Graphics Component (abstract)

## New Actions

Name	Type	Parameters
Check Existence of Entry by Value ( → page 151)	c	String:Text String:Operator Boolean:Exists
Check Selection of Entry by Index/Indices ( → page 152)	c	String:Index/Indices Boolean:Selected
Check Selection of Entry by Value(s) ( → page 153)	c	String:Text String:Operator Boolean:Selected
Drag Entry by Index ( → page 154)	e	Integer:Mouse Button String:Modifier Keys Integer:Index
Drag Entry by Value ( → page 155)	e	Integer:Mouse Button String:Modifier Keys String:Text String:Operator String:Search Type
Drop on Entry by Index ( → page 158)	e	Integer:Index Integer:Delay before drop (milliseconds)
Drop on Entry by Value ( → page 158)	e	String:Text String:Operator String:Search Type Integer:Delay before drop (milliseconds)
Select Entry by Index/Indices ( → page 160)	e	String:Index/Indices String:Extend Selection Integer:Mouse Button
Select Entry by Value(s) ( → page 161)	e	String:Text String:Operator String:Search Type String:Extend Selection Integer:Mouse Button

## Inherited Actions

Name	Type	Parameters	Inherited from
Check Enablement ( → page 45)	c	Boolean:Enabled	Graphics Component (abstract)
Check Enablement of Context Menu Entry by Indxpath ( → page 46)	c	String:Indxpath Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)
Check Enablement of Context Menu Entry by Indxpath (Specify Position) ( → page 47)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Indxpath Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)
Check Enablement of Context Menu Entry by Textpath ( → page 50)	c	String:Textpath String:Operator Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)
Check Enablement of Context Menu Entry by Textpath (Specify Position) ( → page 51)	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 54)	c	Boolean:Exists	Graphics component (ab- stract)	Com- ponent (ab- stract)
Check Existence of Context Menu Entry by Indexpath ( → page 55)	c	String:Indexpath Boolean:Exists Integer:Mouse Button	Graphics component (ab- stract)	Com- ponent (ab- stract)
Check Existence of Context Menu Entry by Indexpath (Specify Position) ( → page 56)	c	Integer:x- position String:x-units Integer:y- position String:y-units String:Indexpath Boolean:Exists Integer:Mouse Button	Graphics component (ab- stract)	Com- ponent (ab- stract)
Check Existence of Context Menu Entry by Textpath ( → page 59)	c	String:Textpath String:Operator Boolean:Exists Integer:Mouse Button	Graphics component (ab- stract)	Com- ponent (ab- stract)
Check Existence of Context Menu Entry by Textpath (Specify Position) ( → page 60)	c	Integer:x- position String:x-units Integer:y- position String:y-units String:Textpath String:Operator Boolean:Exists Integer:Mouse Button	Graphics component (ab- stract)	Com- ponent (ab- stract)
Check Focus ( → page 63)	c	Boolean:Has Fo- cus	Graphics component (ab- stract)	Com- ponent (ab- stract)
Check Property ( → page 64)	c	String:Property Name String:Property Value String:Operator	Graphics component (ab- stract)	Com- ponent (ab- stract)

Check Selection of Context Menu Entry by Indexpath ( → page 65)	c	String:Indexpath Boolean:Selected Integer:Mouse Button	Graphics Component (abstract)	
Check Selection of Context Menu Entry by Indexpath (Specify Position) ( → page 67)	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 69)	c	String:Textpath String:Operator Boolean:Selected Integer:Mouse Button	Graphics Component (abstract)	
Check Selection of Context Menu Entry by Textpath (Specify Position) ( → page 71)	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 32)	c	String:Text String:Operator	Component with Text (abstract)	
Click ( → page 74)	e	Integer:Number of Clicks Integer:Mouse Button	Graphics Component (abstract)	
Click in Component ( → page 75)	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 77)	e	Integer:Mouse Button String:Modifier Keys Integer:x- position String:x-units Integer:y- position String:y-units	Graphics component (ab- stract)	Com- ponent (ab- stract)
Drop ( → page 80)	e	Integer:x- position String:x-units Integer:y- position String:y-units Integer:Delay before drop (milliseconds)	Graphics component (ab- stract)	Com- ponent (ab- stract)
Select Context Menu Entry by Indexpath ( → page 82)	e	String:Indexpath Integer:Mouse Button	Graphics component (ab- stract)	Com- ponent (ab- stract)
Select Context Menu Entry by Indexpath (Specify Position) ( → page 83)	e	Integer:x- position String:x-units Integer:y- position String:y-units String:Indexpath Integer:Mouse Button	Graphics component (ab- stract)	Com- ponent (ab- stract)
Select Context Menu Entry by Textpath ( → page 86)	e	String:Textpath String:Operator Integer:Mouse Button	Graphics component (ab- stract)	Com- ponent (ab- stract)
Select Context Menu Entry by Textpath (Specify Position) ( → page 88)	e	Integer:x- position String:x-units Integer:y- position String:y-units String:Menupath String:Operator Integer:Mouse Button	Graphics component (ab- stract)	Com- ponent (ab- stract)

Show Text ( → page 91)	e	String:Text Integer:Text Size (in points) Integer:Time per Word (in milliseconds) Integer:Window Width (in pixels)	Graphics Component (ab- stract)	Com- ponent (ab- stract)
Store Property ( → page 92)	e	Variable:Variable Name String:Property Name	Graphics Component (ab- stract)	Com- ponent (ab- stract)
Store Value ( → page 33)	e	Variable:Variable Name	Component with Text (ab- stract)	(ab- stract)
Wait for Component ( → page 93)	e	Integer:Timeout in ms Integer:Delay after Visibility	Graphics Component (ab- stract)	Com- ponent (ab- stract)

## 5.2.5.1 Check Existence of Entry by Value

(List)

- 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 15) .
- Select "*matches*" to use a regular expression ( → page 15) .
- Regardless of the operator used: If there is more than one match, the first 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.

### 5.2.5.2 Check Selection of Entry by Index/Indices

(List)

- Use this action to test whether an item or items in a list is/are currently selected.
- The items you want to test are given using their index.

#### Parameters

Name	Data Type	Values	Default
Index/Indices	String	–	none

- Enter the index or indices for the items you want to check.
- For example, if you want to check that the first and third values are selected, enter 1, 3.
- The first item is always 1.
- Separate items using commas ' , ' .

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.

## 5.2.5.3 Check Selection of Entry by Value(s)

(List)

- Use this action to test whether an item or items in a list is/are currently selected.
- The items you want to test are given using their value (name). Each value given is compared (in order) to the corresponding element of the current selection.
- If you only want to check that one value is selected, or to check several selected elements using a single value, you can use the "*check text*" action.
- This checks whether each value you specify is selected in the list.

### Parameters

Name	Data Type	Values	Default
Text	String	–	none

- Enter the name(s) of the items you want to check.
- For example, if you want to check that "*England*" and "*Germany*" are selected, enter *England, Germany*.
- Separate items using commas ' , '.

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 15) .
- Select "*matches*" to use a regular expression ( → page 15) .
- Regardless of the operator used: If there is more than one match, the first found will be chosen.

Name	Data Type	Values	Default
Selected	Boolean	true false	true

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

#### 5.2.5.4 Drag Entry by Index

(List)

- 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.**

## Components, Actions, and 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
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.

### 5.2.5.5 Drag Entry by Value

(List)

- 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 15) 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 15) .
- Select "*matches*" to use a regular expression ( → page 15) .
- Regardless of the operator used: If there is more than one match, the first 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 occurrence of a value, ignoring all elements before and including the selected entry.

### 5.2.5.6 Drop on Entry by Index

(List)

- 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.

### 5.2.5.7 Drop on Entry by Value

(List)

- 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 15) of the item 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 15) .
- Select "*matches*" to use a regular expression ( → page 15) .
- Regardless of the operator used: If there is more than one match, the first 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 occurrence 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.

### 5.2.5.8 Select Entry by Index/Indices

(List)

- Use this action to select an item or items from the list.
- Enter the items you want to select using their indices.
- Jubula realizes this action 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
Index/Indices	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, enter them in plain text.**

---



### 5.2.5.9 Select Entry by Value(s)

(List)

- Use this action to select an item or items from the list.
- Enter the items you want to select using their names (values).
- Jubula realizes this action 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 15) 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 15) .
- Select "*matches*" to use a regular expression ( → page 15) .
- Regardless of the operator used: If there is more than one match, the first 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 occurrence 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

## Components, Actions, and Parameters

---

- 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.**

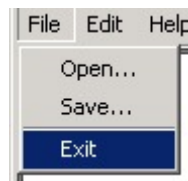
---



## 5.2.6 Menu Bar

### 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 5.10:** 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 379) 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 (concrete)

### New Actions

Name	Type	Parameters
Check Enablement of Entry by Indexpath ( → page 165)	e	String:Indexpath Boolean:Enabled
Check Enablement of Entry by Textpath ( → page 166)	e	String:Menupath String:Operator Boolean:Enabled

Check Existence of Entry by Indxpath ( → page 167)	e	String:Indxpath Boolean:Exists
Check Existence of Entry by Textpath ( → page 168)	e	String:Menupath String:Operator Boolean:Exists
Check Selection of Entry by Indxpath ( → page 169)	e	String:Indxpath Boolean:Selected
Check Selection of Entry by Textpath ( → page 169)	e	String:Menupath String:Operator Boolean:Selected
Select Menu Entry by Indxpath ( → page 171)	e	String:Indxpath
Select Menu Entry by Textpath ( → page 171)	e	String:Menupath String:Operator
Wait for Component ( → page 172)	e	Integer:Timeout in ms Integer:Delay after Visibility

## Inherited Actions

none

### 5.2.6.1 Check Enablement of Entry by Indxpath

(Menu Bar)

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

#### Parameters

Name	Data Type	Values	Default
Indxpath	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 indxpath.
- 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.

### 5.2.6.2 Check Enablement of Entry by Textpath

(Menu Bar)

- 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 15) .

## Components, Actions, and Parameters

- Select "*matches*" to use a regular expression ( → page 15) .
- Regardless of the operator used: If there is more than one match, the first 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.

### 5.2.6.3 Check Existence of Entry by Indexpath

(Menu Bar)

- 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.

#### 5.2.6.4 Check Existence of Entry by Textpath

(Menu Bar)

- 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 15) .
- Select "*matches*" to use a regular expression ( → page 15) .
- Regardless of the operator used: If there is more than one match, the first 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.

## 5.2.6.5 Check Selection of Entry by Indxpath

(Menu Bar)

- 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 indxpath.

### Parameters

Name	Data Type	Values	Default
Indxpath	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 indxpath.
- 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.

## 5.2.6.6 Check Selection of Entry by Textpath

(Menu Bar)

- 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 15) .
- Select "*matches*" to use a regular expression ( → page 15) .
- Regardless of the operator used: If there is more than one match, the first 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.

## 5.2.6.7 Select Menu Entry by Indexpath

(Menu Bar)

- 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)**

---



## 5.2.6.8 Select Menu Entry by Textpath

(Menu Bar)

- 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 15) .
- Select "*matches*" to use a regular expression ( → page 15) .
- Regardless of the operator used: If there is more than one match, the first found will be chosen.

### 5.2.6.9 Wait for Component

(Menu Bar)

- 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) Jubula should wait before the Test Step fails.
- If the menu does not appear in this time frame, the Test Step is unsuccessful.

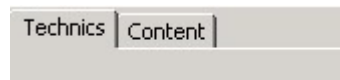
Name	Data Type	Values	Default
Delay after Visibility	Integer	–	200

- 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.

## 5.2.7 Tabbed Pane

### Description:

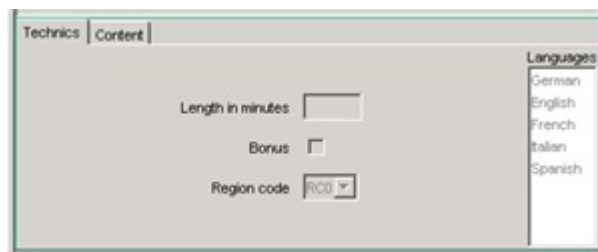
A *tabbed pane* is a container that allows you to switch between multiple groups of components within a single area.



**Figure 5.11:** *Tabbed Pane*

### Mapping tabbed panes

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



**Figure 5.12:** *Tabbed Pane*

### Synopsis:

- Tabbed Pane (concrete)
  - Graphics Component (abstract)

### New Actions

Name	Type	Parameters
Check Enablement of Tab by Index ( → page 179)	c	Integer:Index Boolean:Enabled
Check Enablement of Tab by Value ( → page 180)	c	String:Title String:Operator Boolean:Enabled
Check Existence of Tab ( → page 181)	c	String:Tab String:Operator Boolean:Exists

Check Selection of Tab by Index ( → page 182)	c	Integer:Index Boolean:Selected
Check Selection of Tab by Value ( → page 182)	c	String:Title String:Operator Boolean:Selected
Check Text of Tab by In- dex ( → page 183)	c	Integer:Index String:Title String:Operator
Select Tab by Index ( → page 184)	e	Integer:Index
Select Tab by Value ( → page 185)	e	String:Title String:Operator

## Inherited Actions

Name	Type	Parameters	Inherited from
Check Enablement ( → page 45)	c	Boolean:Enabled	Graphics Component (abstract)
Check Enablement of Context Menu Entry by Indexpath ( → page 46)	c	String:Indexpath Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)
Check Enablement of Context Menu Entry by Indexpath (Specify Position) ( → page 47)	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 50)	c	String:Textpath String:Operator Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)
Check Enablement of Context Menu Entry by Textpath (Specify Position) ( → page 51)	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 54)	c	Boolean:Exists	Graphics Component (abstract)
Check Existence of Context Menu Entry by Indexpath ( → page 55)	c	String:Indexpath Boolean:Exists Integer:Mouse Button	Graphics Component (abstract)

Check Existence of Context Menu Entry by Indexpath (Specify Position) ( → page 56)	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 59)	c	String:Textpath String:Operator Boolean:Exists Integer:Mouse Button	Graphics Component (abstract)	
Check Existence of Context Menu Entry by Textpath (Specify Position) ( → page 60)	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 63)	c	Boolean:Has Focus	Graphics Component (abstract)	
Check Property ( → page 64)	c	String:Property Name String:Property Value String:Operator	Graphics Component (abstract)	
Check Selection of Context Menu Entry by Indexpath ( → page 65)	c	String:Indexpath Boolean:Selected Integer:Mouse Button	Graphics Component (abstract)	
Check Selection of Context Menu Entry by Indexpath (Specify Position) ( → page 67)	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 69)	c	String:Textpath String:Operator Boolean:Selected Integer:Mouse Button	Graphics Component (abstract)	Component (abstract)
Check Selection of Context Menu Entry by Textpath (Specify Position) ( → page 71)	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)	Component (abstract)
Click ( → page 74)	e	Integer:Number of Clicks Integer:Mouse Button	Graphics Component (abstract)	Component (abstract)
Click in Component ( → page 75)	e	Integer:Number of Clicks Integer:Mouse Button Integer:x-position String:x-units Integer:y-position String:y-units	Graphics Component (abstract)	Component (abstract)
Drag ( → page 77)	e	Integer:Mouse Button String:Modifier Keys Integer:x-position String:x-units Integer:y-position String:y-units	Graphics Component (abstract)	Component (abstract)

Drop ( → page 80)	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 82)	e	String:Indexpath Integer:Mouse Button	Graphics Component (abstract)
Select Context Menu Entry by Indexpath (Specify Position) ( → page 83)	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 86)	e	String:Textpath String:Operator Integer:Mouse Button	Graphics Component (abstract)
Select Context Menu Entry by Textpath (Specify Position) ( → page 88)	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 91)	e	String:Text Integer:Text Size (in points) Integer:Time per Word (in milliseconds) Integer:Window Width (in pixels)	Graphics component (ab- stract)	Com- ponent (ab- stract)
Store Property ( → page 92)	e	Variable:Variable Name String:Property Name	Graphics component (ab- stract)	Com- ponent (ab- stract)
Wait for Component ( → page 93)	e	Integer:Timeout in ms Integer:Delay after Visibility	Graphics component (ab- stract)	Com- ponent (ab- stract)

## 5.2.7.1 Check Enablement of Tab by Index

(Tabbed Pane)

- 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.

### 5.2.7.2 Check Enablement of Tab by Value

(Tabbed Pane)

- 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 15) .
- Select "*matches*" to use a regular expression ( → page 15) .
- Regardless of the operator used: If there is more than one match, the first 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.

## 5.2.7.3 Check Existence of Tab

(Tabbed Pane)

- 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 15) .
- Select "*matches*" to use a regular expression ( → page 15) .
- Regardless of the operator used: If there is more than one match, the first 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.

#### 5.2.7.4 Check Selection of Tab by Index

(Tabbed Pane)

- 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.

#### 5.2.7.5 Check Selection of Tab by Value

(Tabbed Pane)

- 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 15) .
- Select "*matches*" to use a regular expression ( → page 15) .
- Regardless of the operator used: If there is more than one match, the first 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.

## 5.2.7.6 Check Text of Tab by Index

(Tabbed Pane)

- 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 15) .
- Select "*matches*" to use a regular expression ( → page 15) .
- Regardless of the operator used: If there is more than one match, the first found will be chosen.

### 5.2.7.7 Select Tab by Index

(Tabbed Pane)

- Use this action to select/activate a tab.
- You choose the tab to select by giving its index.
- Jubula realises this 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.

## 5.2.7.8 Select Tab by Value

(Tabbed Pane)

- Use this action to select/activate a tab.
- You choose the tab to select by giving its title.
- Jubula realises this 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 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 15) .
- Select "*matches*" to use a regular expression ( → page 15) .
- Regardless of the operator used: If there is more than one match, the first found will be chosen.

## 5.2.8 Table

### 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 5.13:** 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 5.14:** Table

### Synopsis:

- Table (concrete)
  - Component with Text Input (abstract)
    - \* Component with Text (abstract)
      - Graphics Component (abstract)

### New Actions

Name	Type	Parameters
Check Editability (Specify Cell) ( → page 195)	c	Boolean:Editable String:Row String:Row Operator String:Column String:Column Operator

Check Editability of Cell (Mouse Position) ( → page 197)	c	Boolean:Editable
Check Editability of Selected Cell ( → page 197)	c	Boolean:Editable
Check Existence of Value in Column ( → page 198)	c	String:Column String:Column Operator String:Value String:Value Operator String:Search Type Boolean:Exists
Check Existence of Value in Row ( → page 200)	c	String:Row String:Row Operator String:Value String:Value Operator String:Search Type Boolean:Exists
Check Text (Mouse Position) ( → page 201)	c	String:Text String:Operator
Check Text (Specify Cell) ( → page 202)	c	String:Text String:Text Operator String:Row String:Row Operator String:Column String:Column Operator
Drag Cell ( → page 204)	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 209)	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 212)	e	Integer:Mouse Button String:Modifier Keys String:Row String:Row Operator String:Value String:Value Operator String:Search Type
Drop on Cell ( → page 215)	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 218)	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 220)	e	String:Row String:Row Operator String:Value String:Value Operator String:Search Type Integer:Delay before drop (milliseconds)

Input Text (Specify Cell) ( → page 222)	e	String:Text String:Row String:Row Operator String:Column String:Column Operator
Move ( → page 224)	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 227)	e	String:Text String:Row String:Row Operator String:Column String:Column Operator
Select Cell ( → page 229)	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 233)	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 236)	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 238)	e	Variable:Variable Name String:Row String:Row Operator String:Column String:Column Operator
Store Value at Mouse Position ( → page 240)	e	Variable:Variable Name

### Inherited Actions

Name	Type	Parameters	Inherited from
Check Editability ( → page 39)	c	Boolean:Editable	Component with Text Input (abstract)
Check Enablement ( → page 45)	c	Boolean:Enabled	Graphics Component (abstract)
Check Enablement of Context Menu Entry by Indxpath ( → page 46)	c	String:Indxpath Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)
Check Enablement of Context Menu Entry by Indxpath (Specify Position) ( → page 47)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Indxpath Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)

Check Enablement of Context Menu Entry by Textpath ( → page 50)	c	String:Textpath String:Operator Boolean:Enabled Integer:Mouse Button	Graphics component (abstract)	Component (abstract)
Check Enablement of Context Menu Entry by Textpath (Specify Position) ( → page 51)	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)	Component (abstract)
Check Existence ( → page 54)	c	Boolean:Exists	Graphics component (abstract)	Component (abstract)
Check Existence of Context Menu Entry by Indexpath ( → page 55)	c	String:Indexpath Boolean:Exists Integer:Mouse Button	Graphics component (abstract)	Component (abstract)
Check Existence of Context Menu Entry by Indexpath (Specify Position) ( → page 56)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Indexpath Boolean:Exists Integer:Mouse Button	Graphics component (abstract)	Component (abstract)
Check Existence of Context Menu Entry by Textpath ( → page 59)	c	String:Textpath String:Operator Boolean:Exists Integer:Mouse Button	Graphics component (abstract)	Component (abstract)
Check Existence of Context Menu Entry by Textpath (Specify Position) ( → page 60)	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)	Component (abstract)

Check Focus ( → page 63)	c	Boolean:Has Focus	Graphics Component (abstract)	Component (abstract)
Check Property ( → page 64)	c	String:PropertyName String:PropertyValue String:Operator	Graphics Component (abstract)	Component (abstract)
Check Selection of Context Menu Entry by Indexpath ( → page 65)	c	String:Indexpath Boolean:Selected Integer:MouseButton	Graphics Component (abstract)	Component (abstract)
Check Selection of Context Menu Entry by Indexpath (Specify Position) ( → page 67)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Indexpath Boolean:Selected Integer:MouseButton	Graphics Component (abstract)	Component (abstract)
Check Selection of Context Menu Entry by Textpath ( → page 69)	c	String:Textpath String:Operator Boolean:Selected Integer:MouseButton	Graphics Component (abstract)	Component (abstract)
Check Selection of Context Menu Entry by Textpath (Specify Position) ( → page 71)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Textpath String:Operator Boolean:Selected Integer:MouseButton	Graphics Component (abstract)	Component (abstract)

Check Text ( → page 32)	c	String:Text String:Operator	Component with Text (ab- stract)	
Click ( → page 74)	e	Integer:Number of Clicks Integer:Mouse Button	Graphics component (ab- stract)	Com- ponent (ab- stract)
Click in Component ( → page 75)	e	Integer:Number of Clicks Integer:Mouse Button Integer:x- position String:x-units Integer:y- position String:y-units	Graphics component (ab- stract)	Com- ponent (ab- stract)
Drag ( → page 77)	e	Integer:Mouse Button String:Modifier Keys Integer:x- position String:x-units Integer:y- position String:y-units	Graphics component (ab- stract)	Com- ponent (ab- stract)
Drop ( → page 80)	e	Integer:x- position String:x-units Integer:y- position String:y-units Integer:Delay before drop (milliseconds)	Graphics component (ab- stract)	Com- ponent (ab- stract)

Input Text ( → page 39)	e	String:Text	Component with Text Input (abstract)
Replace Text ( → page 40)	e	String:Text	Component with Text Input (abstract)
Select Context Menu Entry by Indexpath ( → page 82)	e	String:Indexpath Integer:Mouse Button	Graphics Com- ponent (ab- stract)
Select Context Menu Entry by Indexpath (Specify Position) ( → page 83)	e	Integer:x- position String:x-units Integer:y- position String:y-units String:Indexpath Integer:Mouse Button	Graphics Com- ponent (ab- stract)
Select Context Menu Entry by Textpath ( → page 86)	e	String:Textpath String:Operator Integer:Mouse Button	Graphics Com- ponent (ab- stract)
Select Context Menu Entry by Textpath (Specify Position) ( → page 88)	e	Integer:x- position String:x-units Integer:y- position String:y-units String:Menupath String:Operator Integer:Mouse Button	Graphics Com- ponent (ab- stract)
Show Text ( → page 91)	e	String:Text Integer:Text Size (in points) Integer:Time per Word (in milliseconds) Integer:Window Width (in pixels)	Graphics Com- ponent (ab- stract)

Store Property ( → page 92)	e	Variable:Variable Name String:Property Name	Graphics Component (abstract)	Component (abstract)
Store Value ( → page 33)	e	Variable:Variable Name	Component with Text (abstract)	Component (abstract)
Wait for Component ( → page 93)	e	Integer:Timeout in ms Integer:Delay after Visibility	Graphics Component (abstract)	Component (abstract)

## 5.2.8.1 Check Editability (Specify Cell)

(Table)

- 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 15) .
- Select "*matches*" to use a regular expression ( → page 15) .
- Regardless of the operator used: If there is more than one match, the first 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*.

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.

## Components, Actions, and Parameters

- Select "*simple match*" to use a simple match expression ( → page 15) .
- Select "*matches*" to use a regular expression ( → page 15) .
- Regardless of the operator used: If there is more than one match, the first found will be chosen.

### 5.2.8.2 Check Editability of Cell (Mouse Position)

(Table)

- 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.

### 5.2.8.3 Check Editability of Selected Cell

(Table)

- 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.

## 5.2.8.4 Check Existence of Value in Column

(Table)

- 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*.

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 15) .

## Components, Actions, and Parameters

- Select "*matches*" to use a regular expression ( → page 15) .
- Regardless of the operator used: If there is more than one match, the first 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 15) .
- Select "*matches*" to use a regular expression ( → page 15) .
- Regardless of the operator used: If there is more than one match, the first 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 occurrence 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.

### 5.2.8.5 Check Existence of Value in Row

(Table)

- 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 15) .
- Select "*matches*" to use a regular expression ( → page 15) .
- Regardless of the operator used: If there is more than one match, the first found will be chosen.

Name	Data Type	Values	Default
Value	String	–	none

- Enter the value whose existence you want to check.

## Components, Actions, and Parameters

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 15) .
- Select "*matches*" to use a regular expression ( → page 15) .
- Regardless of the operator used: If there is more than one match, the first 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 occurrence 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.

### 5.2.8.6 Check Text (Mouse Position)

(Table)

- 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 15) .
- Select "*matches*" to use a regular expression ( → page 15) .
- Regardless of the operator used: If there is more than one match, the first found will be chosen.

## 5.2.8.7 Check Text (Specify Cell)

(Table)

- 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 15) .
- Select "*matches*" to use a regular expression ( → page 15) .
- Regardless of the operator used: If there is more than one match, the first 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 15) .
- Select "*matches*" to use a regular expression ( → page 15) .
- Regardless of the operator used: If there is more than one match, the first 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*.

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 15) .
- Select "*matches*" to use a regular expression ( → page 15) .
- Regardless of the operator used: If there is more than one match, the first found will be chosen.

#### 5.2.8.8 Drag Cell

(Table)

- 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, 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
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 15) .
- Select "*matches*" to use a regular expression ( → page 15) .
- Regardless of the operator used: If there is more than one match, the first found will be chosen.

## Components, Actions, and Parameters

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*.

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 15) .
- Select "*matches*" to use a regular expression ( → page 15) .
- Regardless of the operator used: If there is more than one match, the first 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 tell Jubula 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.**

## 5.2.8.9 Drag Cell from Column

(Table)

- 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 **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 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*.

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 15) .
- Select "*matches*" to use a regular expression ( → page 15) .

- Regardless of the operator used: If there is more than one match, the first 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 15) .
- Select "*matches*" to use a regular expression ( → page 15) .
- Regardless of the operator used: If there is more than one match, the first 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 occurrence of a value, ignoring all elements before and including the selected entry.

### 5.2.8.10 Drag Cell from Row

(Table)

- 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

## Components, Actions, and Parameters

- 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 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 15) .
- Select "*matches*" to use a regular expression ( → page 15) .
- Regardless of the operator used: If there is more than one match, the first 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 15) .
- Select "*matches*" to use a regular expression ( → page 15) .
- Regardless of the operator used: If there is more than one match, the first 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 occurrence of a value, ignoring all elements before and including the selected entry.

### 5.2.8.11 Drop on Cell

(Table)

- 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 15) .
- Select "*matches*" to use a regular expression ( → page 15) .
- Regardless of the operator used: If there is more than one match, the first 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*.

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 15) .
- Select "*matches*" to use a regular expression ( → page 15) .
- Regardless of the operator used: If there is more than one match, the first 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 tell Jubula 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 tell Jubula 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.

### 5.2.8.12 Drop on Cell from Column

(Table)

- 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*.

## Components, Actions, and Parameters

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 15) .
- Select "*matches*" to use a regular expression ( → page 15) .
- Regardless of the operator used: If there is more than one match, the first 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 15) .
- Select "*matches*" to use a regular expression ( → page 15) .
- Regardless of the operator used: If there is more than one match, the first 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 occurrence 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.

### 5.2.8.13 Drop on Cell from Row

(Table)

- 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.

## Components, Actions, and Parameters

- 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 15) .
- Select "*matches*" to use a regular expression ( → page 15) .
- Regardless of the operator used: If there is more than one match, the first 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 15) .
- Select "*matches*" to use a regular expression ( → page 15) .
- Regardless of the operator used: If there is more than one match, the first 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 occurrence 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.

#### 5.2.8.14 Input Text (Specify Cell)

(Table)

- Use this action to enter a text into a cell.
- You specify the text to enter and the cell to enter it into. Jubula realizes this by:
  - Clicking once on the cell to activate it
  - If the click does not activate the cell, Jubula double-clicks on the cell
  - Entering the text in the cell.



**This action is unsupported for the HTML toolkit**

#### Parameters

Name	Data Type	Values	Default
Text	String	–	none

## Components, Actions, and Parameters

- 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 15) .
- Select "*matches*" to use a regular expression ( → page 15) .
- Regardless of the operator used: If there is more than one match, the first 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*.

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 15) .
- Select "*matches*" to use a regular expression ( → page 15) .
- Regardless of the operator used: If there is more than one match, the first found will be chosen.

### 5.2.8.15 Move

(Table)

- 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.
- Jubula realises this 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 tell Jubula 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 tell Jubula 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.

### 5.2.8.16 Replace Text (Specify Cell)

(Table)

- Use this action to replace any text already in the cell.
- The text you want to enter is given as a parameter.
- Jubula realizes this 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 15) .
- Select "*matches*" to use a regular expression ( → page 15) .
- Regardless of the operator used: If there is more than one match, the first 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*.

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.

## Components, Actions, and Parameters

- "equals" looks for an exact match.
- Select "simple match" to use a simple match expression ( → page 15) .
- Select "matches" to use a regular expression ( → page 15) .
- Regardless of the operator used: If there is more than one match, the first found will be chosen.

### 5.2.8.17 Select Cell

(Table)

- Use this action to select a cell.
- You specify the cell by giving the row number and column header or index.
- Jubula realises this 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 15) .
- Select "*matches*" to use a regular expression ( → page 15) .
- Regardless of the operator used: If there is more than one match, the first 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*.

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 15) .
- Select "*matches*" to use a regular expression ( → page 15) .
- Regardless of the operator used: If there is more than one match, the first 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) positions.**



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

- Use this parameter to tell Jubula 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 tell Jubula 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, enter them in plain text.**



### 5.2.8.18 Select Value from Column

(Table)

- 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.
- Jubula realizes this 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*.

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 15) .
- Select "*matches*" to use a regular expression ( → page 15) .
- Regardless of the operator used: If there is more than one match, the first 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 15) .
- Select "*matches*" to use a regular expression ( → page 15) .
- Regardless of the operator used: If there is more than one match, the first 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 occurrence 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.**

---



### 5.2.8.19 Select Value from Row

(Table)

- 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.
- Jubula realizes this 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 15) .

- Select "*matches*" to use a regular expression ( → page 15) .
- Regardless of the operator used: If there is more than one match, the first 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 15) .
- Select "*matches*" to use a regular expression ( → page 15) .
- Regardless of the operator used: If there is more than one match, the first 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 occurrence 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.**

### 5.2.8.20 Store Value (Specify Cell)

(Table)

- 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

## Components, Actions, and Parameters

- 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 15) .
- Select "*matches*" to use a regular expression ( → page 15) .
- Regardless of the operator used: If there is more than one match, the first 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*.

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 15) .
- Select "*matches*" to use a regular expression ( → page 15) .
- Regardless of the operator used: If there is more than one match, the first found will be chosen.

### 5.2.8.21 Store Value at Mouse Position

(Table)

- 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

## Components, Actions, and Parameters

---

- Enter the name you want to give to this variable.
- Variable names may only contain letters, numbers and under-scores.
- 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.

## 5.2.9 Text Field/Text Area/Editor Pane/Text Pane

### Description:

This class of components describes generally all editable text areas.



**Figure 5.15:** Text Field

### Mapping text fields

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



**Figure 5.16:** Text Field

### Synopsis:

- Text Field/Text Area/Editor Pane/Text Pane (concrete)
  - Component with Text Input (abstract)
    - \* Component with Text (abstract)
      - Graphics Component (abstract)

### New Actions

Name	Type	Parameters
Insert Text after Index ( → page 247)	e	String:Text Integer:Index/Indices
Insert Text before/after Pattern ( → page 248)	e	String:Text String:Pattern String:Operator Boolean:Insert after Pattern
Select All ( → page 249)	e	
Select Pattern ( → page 249)	e	String:Pattern String:Operator

### Inherited Actions

Name	Type	Parameters	Inherited from
Check Editability ( → page 39)	c	Boolean:Editable	Component with Text Input (abstract)

Check Enablement ( → page 45)	c	Boolean:Enabled	Graphics component (abstract)	Component (abstract)
Check Enablement of Context Menu Entry by Indexpath ( → page 46)	c	String:Indexpath Boolean:Enabled Integer:Mouse Button	Graphics component (abstract)	Component (abstract)
Check Enablement of Context Menu Entry by Indexpath (Specify Position) ( → page 47)	c	Integer:x- position String:x-units Integer:y- position String:y-units String:Indexpath Boolean:Enabled Integer:Mouse Button	Graphics component (abstract)	Component (abstract)
Check Enablement of Context Menu Entry by Textpath ( → page 50)	c	String:Textpath String:Operator Boolean:Enabled Integer:Mouse Button	Graphics component (abstract)	Component (abstract)
Check Enablement of Context Menu Entry by Textpath (Specify Position) ( → page 51)	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)	Component (abstract)
Check Existence ( → page 54)	c	Boolean:Exists	Graphics component (abstract)	Component (abstract)
Check Existence of Context Menu Entry by Indexpath ( → page 55)	c	String:Indexpath Boolean:Exists Integer:Mouse Button	Graphics component (abstract)	Component (abstract)

Check Existence of Context Menu Entry by Indexpath (Specify Position) ( → page 56)	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 59)	c	String:Textpath String:Operator Boolean:Exists Integer:Mouse Button	Graphics Component (abstract)	
Check Existence of Context Menu Entry by Textpath (Specify Position) ( → page 60)	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 63)	c	Boolean:Has Focus	Graphics Component (abstract)	
Check Property ( → page 64)	c	String:Property Name String:Property Value String:Operator	Graphics Component (abstract)	
Check Selection of Context Menu Entry by Indexpath ( → page 65)	c	String:Indexpath Boolean:Selected Integer:Mouse Button	Graphics Component (abstract)	
Check Selection of Context Menu Entry by Indexpath (Specify Position) ( → page 67)	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 69)	c	String:Textpath String:Operator Boolean:Selected Integer:Mouse Button	Graphics Component (abstract)	
Check Selection of Context Menu Entry by Textpath (Specify Position) ( → page 71)	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 32)	c	String:Text String:Operator	Component with Text (abstract)	
Click ( → page 74)	e	Integer:Number of Clicks Integer:Mouse Button	Graphics Component (abstract)	
Click in Component ( → page 75)	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 77)	e	Integer:Mouse Button String:Modifier Keys Integer:x-position String:x-units Integer:y-position String:y-units	Graphics Component (abstract)	

Drop ( → page 80)	e	Integer:x-position String:x-units Integer:y-position String:y-units Integer:Delay before drop (milliseconds)	Graphics Component (abstract)
Input Text ( → page 39)	e	String:Text	Component with Text Input (abstract)
Replace Text ( → page 40)	e	String:Text	Component with Text Input (abstract)
Select Context Menu Entry by Indexpath ( → page 82)	e	String:Indexpath Integer:Mouse Button	Graphics Component (abstract)
Select Context Menu Entry by Indexpath (Specify Position) ( → page 83)	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 86)	e	String:Textpath String:Operator Integer:Mouse Button	Graphics Component (abstract)
Select Context Menu Entry by Textpath (Specify Position) ( → page 88)	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 91)	e	String:Text Integer:Text Size (in points) Integer:Time per Word (in milliseconds) Integer:Window Width (in pixels)	Graphics Component (abstract)	Component (abstract)
Store Property ( → page 92)	e	Variable:Variable Name String:Property Name	Graphics Component (abstract)	Component (abstract)
Store Value ( → page 33)	e	Variable:Variable Name	Component with Text (abstract)	Component (abstract)
Wait for Component ( → page 93)	e	Integer:Timeout in ms Integer:Delay after Visibility	Graphics Component (abstract)	Component (abstract)

## 5.2.9.1 Insert Text after Index

(Text Field/Text Area/Editor Pane/Text Pane)

- 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.
- Jubula realizes this 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/Indices	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.

### 5.2.9.2 Insert Text before/after Pattern

(Text Field/Text Area/Editor Pane/Text Pane)

- 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.
- Jubula realizes this 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.

## Components, Actions, and Parameters

- "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 15) .
- Select "matches" to use a regular expression ( → page 15) .
- Regardless of the operator used: If there is more than one match, the first 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.

### 5.2.9.3 Select All

(Text Field/Text Area/Editor Pane/Text Pane)

- This action selects the entire text within the component.
- Jubula realizes this by clicking three times within the text field.

---

**This action is unsupported for the HTML toolkit**

---



This action has no parameters.

### 5.2.9.4 Select Pattern

(Text Field/Text Area/Editor Pane/Text Pane)

- Use this action to select a given pattern within the text.
- If the pattern is not found, no text is selected.
- Jubula realizes this 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 15) .
- Select "*matches*" to use a regular expression ( → page 15) .
- Regardless of the operator used: If there is more than one match, the first found will be chosen.

## 5.2.10 Tree

### 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*).

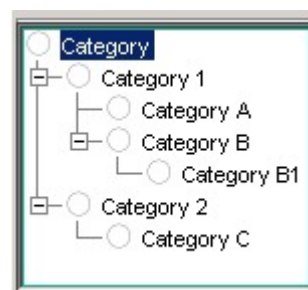


**Figure 5.17:** Tree

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 379) for more details.

### Mapping trees

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



**Figure 5.18:** Tree

**Actions on trees (as a hierarchical component) are not supported in the HTML toolkit. Individual nodes must be addressed as single links.**



### Synopsis:

- Tree (concrete)
  - Graphics Component (abstract)

## New Actions

Name	Type	Parameters
Check Existence of Node by Indexpath ( → page 258)	c	String:Path Type Integer:Pre-ascend String:Indexpath Boolean:Exists
Check Existence of Node by Textpath ( → page 259)	c	String:Path Type Integer:Pre-ascend String:Textpath String:Operator Boolean:Exists
Check Text (Mouse Position) ( → page 261)	c	String:Text String:Operator
Check Text of Selected Node(s) ( → page 262)	c	String:Text String:Operator
Collapse Node by Indexpath ( → page 263)	e	String:Path Type Integer:Pre-ascend String:Indexpath
Collapse Node by Textpath ( → page 265)	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 270)	e	Integer:Mouse Button String:Modifier Keys String:Path Type Integer:Pre-ascend String:Tree Textpath String:Operator
Drop on Node by Indexpath ( → page 273)	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 Index- path ( → page 276)	e	String:Path Type Integer:Pre-ascend String:Indexpath
Expand Node by Textpath ( → page 278)	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 Index- path ( → page 282)	e	String:Path Type Integer:Pre-ascend String:Indexpath Integer:Number of Clicks Integer:Mouse Button String:Extend Selection
Select Node by Textpath ( → page 284)	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 Po- sition ( → page 287)	e	Variable:Variable Name
Store Text of Selected Node ( → page 288)	e	Variable:Variable Name

## Inherited Actions

Name	Type	Parameters	Inherited from
Check Enablement ( → page 45)	c	Boolean:Enabled	Graphics Component (abstract)
Check Enablement of Context Menu Entry by Indexpath ( → page 46)	c	String:Indexpath Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)

Check Enablement of Context Menu Entry by Indexpath (Specify Position) ( → page 47)	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 50)	c	String:Textpath String:Operator Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)	
Check Enablement of Context Menu Entry by Textpath (Specify Position) ( → page 51)	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 54)	c	Boolean:Exists	Graphics Component (abstract)	
Check Existence of Context Menu Entry by Indexpath ( → page 55)	c	String:Indexpath Boolean:Exists Integer:Mouse Button	Graphics Component (abstract)	
Check Existence of Context Menu Entry by Indexpath (Specify Position) ( → page 56)	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 59)	c	String:Textpath String:Operator Boolean:Exists Integer:Mouse Button	Graphics Component (abstract)	
Check Existence of Context Menu Entry by Textpath (Specify Position) ( → page 60)	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 63)	c	Boolean:Has Focus	Graphics Component (abstract)	
Check Property ( → page 64)	c	String:Property Name String:Property Value String:Operator	Graphics Component (abstract)	
Check Selection of Context Menu Entry by Indexpath ( → page 65)	c	String:Indexpath Boolean:Selected Integer:Mouse Button	Graphics Component (abstract)	
Check Selection of Context Menu Entry by Indexpath (Specify Position) ( → page 67)	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 69)	c	String:Textpath String:Operator Boolean:Selected Integer:Mouse Button	Graphics Component (abstract)
Check Selection of Context Menu Entry by Textpath (Specify Position) ( → page 71)	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 74)	e	Integer:Number of Clicks Integer:Mouse Button	Graphics Component (abstract)
Click in Component ( → page 75)	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 77)	e	Integer:Mouse Button String:Modifier Keys Integer:x-position String:x-units Integer:y-position String:y-units	Graphics Component (abstract)

Drop ( → page 80)	e	Integer:x-position String:x-units Integer:y-position String:y-units Integer:Delay before drop (milliseconds)	Graphics component (abstract)	Component (abstract)
Select Context Menu Entry by Indexpath ( → page 82)	e	String:Indexpath Integer:Mouse Button	Graphics component (abstract)	Component (abstract)
Select Context Menu Entry by Indexpath (Specify Position) ( → page 83)	e	Integer:x-position String:x-units Integer:y-position String:y-units String:Indexpath Integer:Mouse Button	Graphics component (abstract)	Component (abstract)
Select Context Menu Entry by Textpath ( → page 86)	e	String:Textpath String:Operator Integer:Mouse Button	Graphics component (abstract)	Component (abstract)
Select Context Menu Entry by Textpath (Specify Position) ( → page 88)	e	Integer:x-position String:x-units Integer:y-position String:y-units String:Menupath String:Operator Integer:Mouse Button	Graphics component (abstract)	Component (abstract)

Show Text ( → page 91)	e	String:Text Integer:Text Size (in points) Integer:Time per Word (in milliseconds) Integer:Window Width (in pixels)	Graphics Component (ab- stract)	Com- ponent (ab- stract)
Store Property ( → page 92)	e	Variable:Variable Name String:Property Name	Graphics Component (ab- stract)	Com- ponent (ab- stract)
Wait for Component ( → page 93)	e	Integer:Timeout in ms Integer:Delay after Visibility	Graphics Component (ab- stract)	Com- ponent (ab- stract)

### 5.2.10.1 Check Existence of Node by Indxpath

(Tree)

- This action checks that a node in the tree exists.
- You give the path to the node you want to check as an indxpath.
- 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

## Components, Actions, and Parameters

- 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
Indxpath	String	–	none

Enter the indxpath 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 indxpath.
- 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 indxpath to exist.
- Set to false if you expect the indxpath *not* to exist.

### 5.2.10.2 Check Existence of Node by Textpath

(Tree)

- 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.

## Components, Actions, and Parameters

- 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 15) .
- Select "*matches*" to use a regular expression ( → page 15) .
- Regardless of the operator used: If there is more than one match, the first 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.

### 5.2.10.3 Check Text (Mouse Position)

(Tree)

- 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 15) .
- Select "*matches*" to use a regular expression ( → page 15) .
- Regardless of the operator used: If there is more than one match, the first found will be chosen.

### 5.2.10.4 Check Text of Selected Node(s)

(Tree)

- 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 15) ) you want to check.
- If you want to select multiple values, separate the values with a comma ', '.
- This parameter will be compared to the selected nodes, using the operator provided.

## Components, Actions, and Parameters

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 15) .
- Select "*matches*" to use a regular expression ( → page 15) .
- Regardless of the operator used: If there is more than one match, the first found will be chosen.

### 5.2.10.5 Collapse Node by Indexpath

(Tree)

- Use this action to collapse a tree or part of a tree along an indicated path.
- You enter the path as an indexpath.
- Jubula realises this 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).

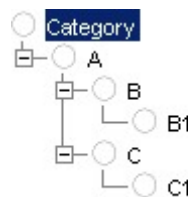
- Enter the path to the item as an indexpath.
- Use slash '/' as a path separator (to separate parent nodes from child nodes).
- For example, 1/2 (without quotes).



**The first node is '1' (without quotes)**

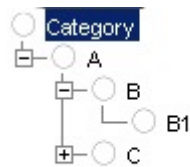
#### Example:

- Your tree looks like this:



**Figure 5.19:** Tree 1

- You want to collapse node "C".
- Enter 1/1/2:
- To collapse node "A", enter 1/1:
- To collapse a whole tree, enter 1.



**Figure 5.20:** Tree 2



**Figure 5.21:** Tree 3

## 5.2.10.6 Collapse Node by Textpath

(Tree)

- Use this action to collapse a tree or part of a tree along an indicated path.
- You enter the path as a textpath.
- Jubula realises this 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*.**

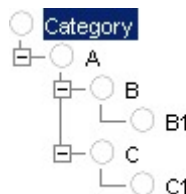
Name	Data Type	Values	Default
Textpath	String	–	none

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.

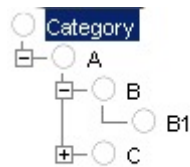
**Example:**

- Your tree looks like this:



**Figure 5.22:** Tree 1

- You want to collapse node "C".
- Enter `Category/A/C`:
- To collapse node "A", enter `Category/A`:



**Figure 5.23:** Tree 2



**Figure 5.24:** 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 15) .
- Select "*matches*" to use a regular expression ( → page 15) .
- Regardless of the operator used: If there is more than one match, the first found will be chosen.

**When you use a regular expression for a tree path, 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.**



## 5.2.10.7 Drag Node by Indexpath

(Tree)

- This action drags the node you specify.

- You give the path to the node to be dragged using an `indexPath`.
- Jubula realizes this by:
  - Moving the cursor over the component.
  - If the tree is not expanded, Jubula expands it.
  - 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 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)**

---

### 5.2.10.8 Drag Node by Textpath

(Tree)

- This action drags the node you specify.
- You give the path to the node to be dragged using a textpath.
- Jubula realizes this by:
  - Moving the cursor over the component.
  - If the tree is not expanded, Jubula expands it.
  - 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 15) .

- Select "*matches*" to use a regular expression ( → page 15) .
- Regardless of the operator used: If there is more than one match, the first found will be chosen.

### 5.2.10.9 Drop on Node by Indxpath

(Tree)

- 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 indxpath.
- Jubula realizes this by:
  - Moving the cursor over the node.
  - If the tree is not expanded, Jubula expands it.
  - 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.

#### 5.2.10.10 Drop on Node by Textpath

(Tree)

- 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.
- Jubula realizes this by:

- Moving the cursor over the node.
  - If the tree is not expanded, Jubula expands it.
  - 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 15) .
- Select "*matches*" to use a regular expression ( → page 15) .
- Regardless of the operator used: If there is more than one match, the first 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.

### 5.2.10.11 Expand Node by Indxpath

(Tree)

- Use this action to expand a tree or part of a tree along an indicated path.
- You enter the path as an *indxpath*.
- Jubula realises this 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)**

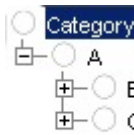




**Figure 5.25:** Tree 3

**Example:**

- Your tree looks like this:
- You want to expand the tree to node A
- Enter 1 / 1:



**Figure 5.26:** Tree 4

## 5.2.10.12 Expand Node by Textpath

(Tree)

- Use this action to expand a tree or part of a tree along an indicated path.
- You enter the path as a textpath.
- Jubula realises this 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).

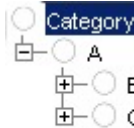
- Enter the path to the item as a textpath.
- Use slash '/' as a path separator (to separate parent nodes from child nodes).
- For example, `File/Open` or `Category/Horror` (without quotes).
- Either make sure that your path is written exactly as it appears in the interface, or use a regular expression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



**Figure 5.27:** Tree 3



**Figure 5.28:** 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 15) .
- Select "*matches*" to use a regular expression ( → page 15) .
- Regardless of the operator used: If there is more than one match, the first 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.**

### 5.2.10.13 Move

(Tree)

- Use this action to move from a currently selected node to another node.

## Components, Actions, and Parameters

- 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.
- Jubula realises this 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

- Use this parameter to define the direction you want to move in.
- Combined with the node count, you can move any number of nodes in any direction.
- The following directions can be used:
  - "next" searches only through sibling nodes (nodes that share a parent with the currently selected node). This search begins with the sibling node directly below the currently selected node and continues downward. If the search proceeds past the bottom sibling, the component is not found.
  - "up" progressively searches parent nodes. This means that the first node found is the parent of the currently selected node. The second node found is the parent of that node, and so on. If the search proceeds beyond the root node, the component is not found.
  - "down" progressively searches first children nodes. This means that the first node found is the top child of the currently selected node. The second node found is the top child of that node, and so on. If the search proceeds beyond a node that has no children, the component is not found.
  - "previous" searches only through sibling nodes (nodes that share a parent with the currently selected node). This search begins with the sibling node directly above the currently selected node and continues upward. If the search proceeds past the top sibling, the component is not found.

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.

#### 5.2.10.14 Select Node by Indexpath

(Tree)

- Use this item to select a node from a tree.
- You give the path to the node using an indexpath.
- Jubula realises this by:
  - Searching for the node you specify.
  - If the tree is not expanded, Jubula expands it.
  - 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.

## Components, Actions, and Parameters

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.

### 5.2.10.15 Select Node by Textpath

(Tree)

- Use this item to select a node from a tree.
- You give the path to the node using a textpath.
- Jubula realises this by:
  - Searching for the node you specify.
  - If the tree is not expanded, Jubula expands it.
  - 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 15) .
- Select "*matches*" to use a regular expression ( → page 15) .
- Regardless of the operator used: If there is more than one match, the first 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, 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.

### 5.2.10.16 Store Text at Mouse Position

(Tree)

- 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.

### 5.2.10.17 Store Text of Selected Node

(Tree)

- 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 under-scores.
- 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)  
Tree Table (swt)

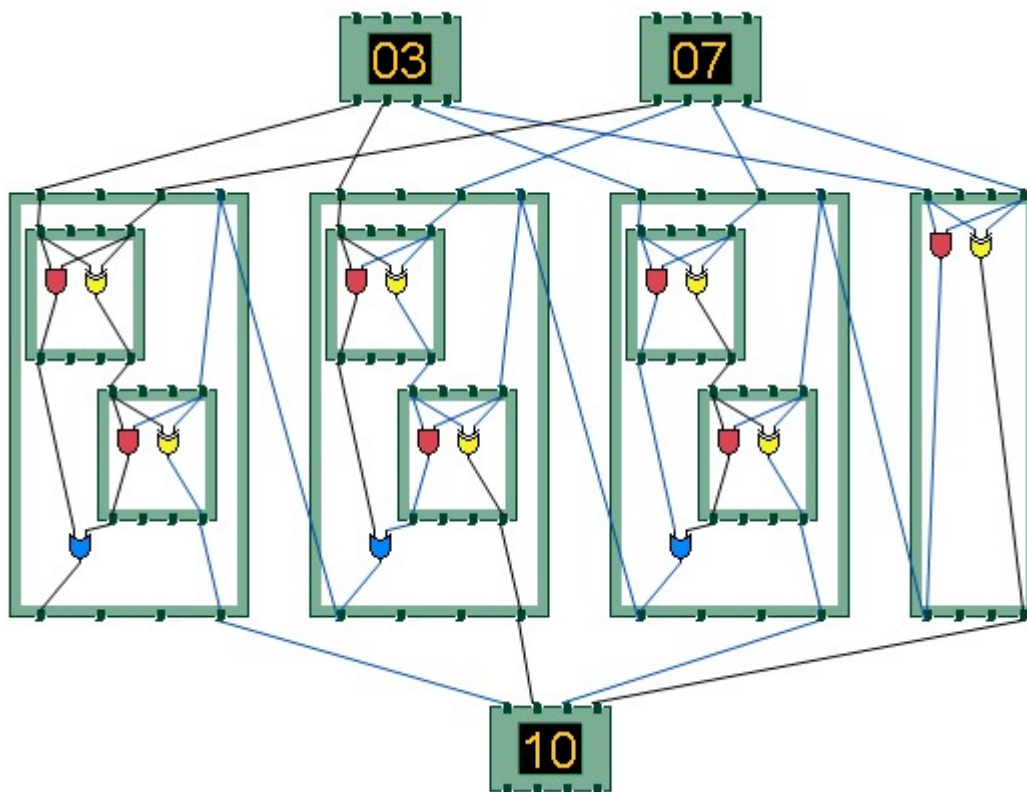
## 5.3 gef Toolkit

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.

### 5.3.1 Figure Canvas

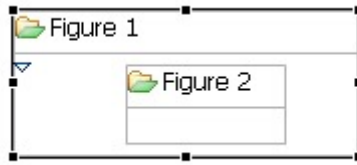
**Description:**

- The figure canvas (Figure 5.29 → page 289 ) is an editor where figures, connections and anchors are displayed.



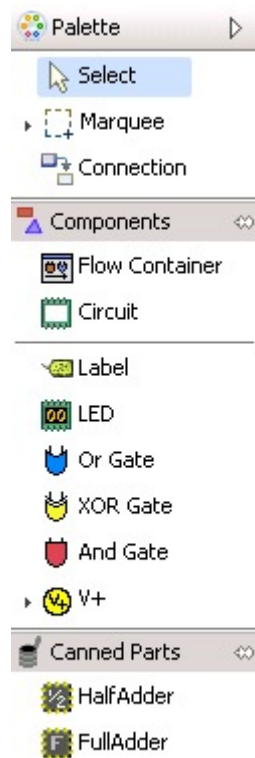
**Figure 5.29:** Figure Canvas

- Testing GEF components in Jubula 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. 213).
- The figure canvas itself contains various items which can be addressed by Jubula.
- Individual figures (Figure 5.30 → page 290 ).
- Tools (Figure 5.31 → page 291 ).
- Connections (Figure 5.32 → page 291 ).

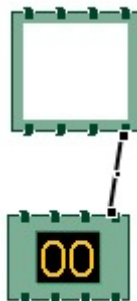


**Figure 5.30:** *A selected figure containing another figure*

- Connection anchors. These are the points where connections join to figures. In Jubula, connection anchors are considered as figures. As such, they can be addressed (clicked, checked etc) using a textpath.



**Figure 5.31:** Various tools on the palette



**Figure 5.32:** A connection between two figures

## Synopsis:

- Figure Canvas (gef)
  - Graphics Component (abstract)

## New Actions

Name	Type	Parameters
Check Figure Existence ( → page 297)	e	String:Textpath String:Operator Boolean:Exists
Check Figure Property ( → page 298)	c	String:Textpath String:Text Operator String:Property Name String:Property Value String:Value Operator
Check Tool Existence ( → page 300)	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 304)	e	String:Textpath String:Operator Integer:Number of Clicks Integer:Mouse Button
Click in Figure ( → page 305)	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 308)	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 314)	e	String:Textpath String:Operator Integer:Number of Clicks

## Inherited Actions

Name	Type	Parameters	Inherited from
Check Enablement ( → page 45)	c	Boolean:Enabled	Graphics Component (abstract)
Check Enablement of Context Menu Entry by Indxpath ( → page 46)	c	String:Indxpath Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)
Check Enablement of Context Menu Entry by Indxpath (Specify Position) ( → page 47)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Indxpath Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)

Check Enablement of Context Menu Entry by Textpath ( → page 50)	c	String:Textpath String:Operator Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)	Component (abstract)
Check Enablement of Context Menu Entry by Textpath (Specify Position) ( → page 51)	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)	Component (abstract)
Check Existence ( → page 54)	c	Boolean:Exists	Graphics Component (abstract)	Component (abstract)
Check Existence of Context Menu Entry by Indxpath ( → page 55)	c	String:Indxpath Boolean:Exists Integer:Mouse Button	Graphics Component (abstract)	Component (abstract)
Check Existence of Context Menu Entry by Indxpath (Specify Position) ( → page 56)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Indxpath Boolean:Exists Integer:Mouse Button	Graphics Component (abstract)	Component (abstract)
Check Existence of Context Menu Entry by Textpath ( → page 59)	c	String:Textpath String:Operator Boolean:Exists Integer:Mouse Button	Graphics Component (abstract)	Component (abstract)
Check Existence of Context Menu Entry by Textpath (Specify Position) ( → page 60)	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)	Component (abstract)

Check Focus ( → page 63)	c	Boolean:Has Focus	Graphics Component (abstract)	Component (abstract)
Check Property ( → page 64)	c	String:PropertyName String:PropertyValue String:Operator	Graphics Component (abstract)	Component (abstract)
Check Selection of Context Menu Entry by Indexpath ( → page 65)	c	String:Indexpath Boolean:Selected Integer:Mouse Button	Graphics Component (abstract)	Component (abstract)
Check Selection of Context Menu Entry by Indexpath (Specify Position) ( → page 67)	c	Integer:x- position String:x-units Integer:y- position String:y-units String:Indexpath Boolean:Selected Integer:Mouse Button	Graphics Component (abstract)	Component (abstract)
Check Selection of Context Menu Entry by Textpath ( → page 69)	c	String:Textpath String:Operator Boolean:Selected Integer:Mouse Button	Graphics Component (abstract)	Component (abstract)
Check Selection of Context Menu Entry by Textpath (Specify Position) ( → page 71)	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)	Component (abstract)

Click ( → page 74)	e	Integer:Number of Clicks Integer:Mouse Button	Graphics component (abstract)	Component (abstract)
Click in Component ( → page 75)	e	Integer:Number of Clicks Integer:Mouse Button Integer:x-position String:x-units Integer:y-position String:y-units	Graphics component (abstract)	Component (abstract)
Drag ( → page 77)	e	Integer:Mouse Button String:Modifier Keys Integer:x-position String:x-units Integer:y-position String:y-units	Graphics component (abstract)	Component (abstract)
Drop ( → page 80)	e	Integer:x-position String:x-units Integer:y-position String:y-units Integer:Delay before drop (milliseconds)	Graphics component (abstract)	Component (abstract)
Select Context Menu Entry by Indexpath ( → page 82)	e	String:Indexpath Integer:Mouse Button	Graphics component (abstract)	Component (abstract)
Select Context Menu Entry by Indexpath (Specify Position) ( → page 83)	e	Integer:x-position String:x-units Integer:y-position String:y-units String:Indexpath Integer:Mouse Button	Graphics component (abstract)	Component (abstract)

Select Context Menu Entry by Textpath ( → page 86)	e	String:Textpath String:Operator Integer:Mouse Button	Graphics Component (abstract)	Component (abstract)
Select Context Menu Entry by Textpath (Specify Position) ( → page 88)	e	Integer:x-position String:x-units Integer:y-position String:y-units String:Menupath String:Operator Integer:Mouse Button	Graphics Component (abstract)	Component (abstract)
Show Text ( → page 91)	e	String:Text Integer:Text Size (in points) Integer:Time per Word (in milliseconds) Integer:Window Width (in pixels)	Graphics Component (abstract)	Component (abstract)
Store Property ( → page 92)	e	Variable:Variable Name String:Property Name	Graphics Component (abstract)	Component (abstract)
Wait for Component ( → page 93)	e	Integer:Timeout in ms Integer:Delay after Visibility	Graphics Component (abstract)	Component (abstract)

## 5.3.1.1 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 15) .
- Select "*matches*" to use a regular expression ( → page 15) .
- Regardless of the operator used: If there is more than one match, the first 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.

### 5.3.1.2 Check Figure Property

(Figure Canvas)

- Use this action to check a specific property of a figure.



**If you want to 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 Jubula 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. 213).

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 15) .
- Select "*matches*" to use a regular expression ( → page 15) .
- Regardless of the operator used: If there is more than one match, the first 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 parameter 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 15) .
- Select "*matches*" to use a regular expression ( → page 15) .
- Regardless of the operator used: If there is more than one match, the first found will be chosen.

### 5.3.1.3 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 parameter to enter the textpath to the tool you want to check.

- Enter the path to the item as a textpath.

## Components, Actions, and Parameters

- 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 15) .
- Select "*matches*" to use a regular expression ( → page 15) .
- Regardless of the operator used: If there is more than one match, the first 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.

### 5.3.1.4 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. 213).

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 15) .
- Select "*matches*" to use a regular expression ( → page 15) .
- Regardless of the operator used: If there is more than one match, the first 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. 213).

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

## Components, Actions, and Parameters

- 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 15) .
- Select "*matches*" to use a regular expression ( → page 15) .
- Regardless of the operator used: If there is more than one match, the first 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.**



### 5.3.1.5 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 15) .
- Select "*matches*" to use a regular expression ( → page 15) .
- Regardless of the operator used: If there is more than one match, the first 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.

## Components, Actions, and 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.**



### 5.3.1.6 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 15) .
- Select "*matches*" to use a regular expression ( → page 15) .
- Regardless of the operator used: If there is more than one match, the first 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) positions.**



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

- Use this parameter to tell Jubula 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 tell Jubula 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.

### 5.3.1.7 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. 213).

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

- The *operator* parameter has four possible values.

## Components, Actions, and Parameters

- "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 15) .
- Select "matches" to use a regular expression ( → page 15) .
- Regardless of the operator used: If there is more than one match, the first 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 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 tell Jubula 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 tell Jubula 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.

### 5.3.1.8 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. 213).

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 15) .
- Select "*matches*" to use a regular expression ( → page 15) .
- Regardless of the operator used: If there is more than one match, the first 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 tell Jubula 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 tell Jubula 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.

### 5.3.1.9 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. 213).

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 15) .
- Select "*matches*" to use a regular expression ( → page 15) .
- Regardless of the operator used: If there is more than one match, the first 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.

## 5.4 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

## 5.4.1 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 316)	e	String:URL
Open URL ( → page 316)	e	
Reload page ( → page 317)	e	

### Inherited Actions

*none*

#### 5.4.1.1 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.

#### 5.4.1.2 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'`.

### 5.4.1.3 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.

## 5.4.2 HTML Hyperlink

### Description:

A HTML hyperlink is a link which can be followed from within a browser.

### Synopsis:

- HTML Hyperlink (html)
  - Component with Text (abstract)
  - \* Graphics Component (abstract)

### New Actions

Name	Type	Parameters
Check URL ( → page 323)	c	String:Text String:Operator

### Inherited Actions

Name	Type	Parameters	Inherited from
Check Enablement ( → page 45)	c	Boolean:Enabled	Graphics Component (abstract)
Check Enablement of Context Menu Entry by Indxpath ( → page 46)	c	String:Indxpath Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)
Check Enablement of Context Menu Entry by Indxpath (Specify Position) ( → page 47)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Indxpath Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)
Check Enablement of Context Menu Entry by Textpath ( → page 50)	c	String:Textpath String:Operator Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)

Check Enablement of Context Menu Entry by Textpath (Specify Position) ( → page 51)	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)	Component (abstract)
Check Existence ( → page 54)	c	Boolean:Exists	Graphics Component (abstract)	Component (abstract)
Check Existence of Context Menu Entry by Indxpath ( → page 55)	c	String:Indxpath Boolean:Exists Integer:Mouse Button	Graphics Component (abstract)	Component (abstract)
Check Existence of Context Menu Entry by Indxpath (Specify Position) ( → page 56)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Indxpath Boolean:Exists Integer:Mouse Button	Graphics Component (abstract)	Component (abstract)
Check Existence of Context Menu Entry by Textpath ( → page 59)	c	String:Textpath String:Operator Boolean:Exists Integer:Mouse Button	Graphics Component (abstract)	Component (abstract)
Check Existence of Context Menu Entry by Textpath (Specify Position) ( → page 60)	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)	Component (abstract)

Check Focus ( → page 63)	c	Boolean:Has Focus	Graphics Component (abstract)	Component (abstract)
Check Property ( → page 64)	c	String:PropertyName String:PropertyValue String:Operator	Graphics Component (abstract)	Component (abstract)
Check Selection of Context Menu Entry by Indexpath ( → page 65)	c	String:Indexpath Boolean:Selected Integer:MouseButton	Graphics Component (abstract)	Component (abstract)
Check Selection of Context Menu Entry by Indexpath (Specify Position) ( → page 67)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Indexpath Boolean:Selected Integer:MouseButton	Graphics Component (abstract)	Component (abstract)
Check Selection of Context Menu Entry by Textpath ( → page 69)	c	String:Textpath String:Operator Boolean:Selected Integer:MouseButton	Graphics Component (abstract)	Component (abstract)
Check Selection of Context Menu Entry by Textpath (Specify Position) ( → page 71)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Textpath String:Operator Boolean:Selected Integer:MouseButton	Graphics Component (abstract)	Component (abstract)

Check Text ( → page 32)	c	String:Text String:Operator	Component with Text (ab- stract)	
Click ( → page 74)	e	Integer:Number of Clicks Integer:Mouse Button	Graphics component (ab- stract)	Com- ponent (ab- stract)
Click in Component ( → page 75)	e	Integer:Number of Clicks Integer:Mouse Button Integer:x- position String:x-units Integer:y- position String:y-units	Graphics component (ab- stract)	Com- ponent (ab- stract)
Drag ( → page 77)	e	Integer:Mouse Button String:Modifier Keys Integer:x- position String:x-units Integer:y- position String:y-units	Graphics component (ab- stract)	Com- ponent (ab- stract)
Drop ( → page 80)	e	Integer:x- position String:x-units Integer:y- position String:y-units Integer:Delay before drop (milliseconds)	Graphics component (ab- stract)	Com- ponent (ab- stract)

Select Context Menu Entry by Indexpath ( → page 82)	e	String:Indexpath Integer:Mouse Button	Graphics Component (abstract)
Select Context Menu Entry by Indexpath (Specify Position) ( → page 83)	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 86)	e	String:Textpath String:Operator Integer:Mouse Button	Graphics Component (abstract)
Select Context Menu Entry by Textpath (Specify Position) ( → page 88)	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 91)	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 92)	e	Variable:Variable Name String:Property Name	Graphics Component (abstract)
Store Value ( → page 33)	e	Variable:Variable Name	Component with Text (abstract)

Wait for Component ( → page 93)	e	Integer:Timeout in ms Integer:Delay after Visibility	Graphics Component (abstract)	Component (abstract)
------------------------------------	---	---	----------------------------------	-------------------------

## 5.4.2.1 Check URL

(HTML Hyperlink)

- 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 15) .
- Select "*matches*" to use a regular expression ( → page 15) .
- Regardless of the operator used: If there is more than one match, the first found will be chosen.

## **5.5 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.

## **5.6 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.

## **5.7 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.

## 5.7.1 (SWT) Tree

### Description:

A tree in SWT is essentially the same as a tree in Swing, with an extra option to have checkboxes 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 (concrete)
  - \* Graphics Component (abstract)

### New Actions

Name	Type	Parameters
Check Selection of Checkbox by Indexpath ( → page 332)	c	String:Path Type Integer:Pre-ascend String:Indexpath Boolean:Checked
Check Selection of Checkbox by Textpath ( → page 334)	c	String:Path Type Integer:Pre-ascend String:Textpath String:Operator Boolean:Checked
Check Selection of Checkbox on Selected Node ( → page 336)	c	Boolean:Checked
Toggle Checkbox on Node by Indexpath ( → page 336)	e	String:Path Type Integer:Pre-ascend String:Indexpath
Toggle Checkbox on Node by Textpath ( → page 337)	e	String:Path Type Integer:Pre-ascend String:Textpath String:Operator

### Inherited Actions

Name	Type	Parameters	Inherited from
Check Enablement ( → page 45)	c	Boolean:Enabled	Graphics Component (abstract)
Check Enablement of Context Menu Entry by Indexpath ( → page 46)	c	String:Indexpath Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)

Check Enablement of Context Menu Entry by Indexpath (Specify Position) ( → page 47)	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 50)	c	String:Textpath String:Operator Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)	
Check Enablement of Context Menu Entry by Textpath (Specify Position) ( → page 51)	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 54)	c	Boolean:Exists	Graphics Component (abstract)	
Check Existence of Context Menu Entry by Indexpath ( → page 55)	c	String:Indexpath Boolean:Exists Integer:Mouse Button	Graphics Component (abstract)	
Check Existence of Context Menu Entry by Indexpath (Specify Position) ( → page 56)	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 59)	c	String:Textpath String:Operator Boolean:Exists Integer:Mouse Button	Graphics Component (abstract)	
Check Existence of Context Menu Entry by Textpath (Specify Position) ( → page 60)	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 258)	c	String:Path Type Integer:Pre-ascend String:Indexpath Boolean:Exists	Tree (concrete)	
Check Existence of Node by Textpath ( → page 259)	c	String:Path Type Integer:Pre-ascend String:Textpath String:Operator Boolean:Exists	Tree (concrete)	
Check Focus ( → page 63)	c	Boolean:Has Focus	Graphics Component (abstract)	
Check Property ( → page 64)	c	String:Property Name String:Property Value String:Operator	Graphics Component (abstract)	
Check Selection of Context Menu Entry by Indexpath ( → page 65)	c	String:Indexpath Boolean:Selected Integer:Mouse Button	Graphics Component (abstract)	

Check Selection of Context Menu Entry by Indexpath (Specify Position) ( → page 67)	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 69)	c	String:Textpath String:Operator Boolean:Selected Integer:Mouse Button	Graphics Component (abstract)
Check Selection of Context Menu Entry by Textpath (Specify Position) ( → page 71)	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 261)	c	String:Text String:Operator	Tree (concrete)
Check Text of Selected Node(s) ( → page 262)	c	String:Text String:Operator	Tree (concrete)
Click ( → page 74)	e	Integer:Number of Clicks Integer:Mouse Button	Graphics Component (abstract)
Click in Component ( → page 75)	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 Ind- expath ( → page 263)	e	String:Path Type Integer:Pre- ascend String:Indexpath	Tree (concrete)
Collapse Node by Textpath ( → page 265)	e	String:Path Type Integer:Pre- ascend String:Textpath String:Operator	Tree (concrete)
Drag ( → page 77)	e	Integer:Mouse Button String:Modifier Keys Integer:x- position String:x-units Integer:y- position String:y-units	Graphics Com- ponent (ab- stract)
Drag Node by Index- path ( → page 267)	e	Integer:Mouse Button String:Modifier Keys String:Path Type Integer:Pre- ascend String:Tree Ind- expath	Tree (concrete)
Drag Node by Textpath ( → page 270)	e	Integer:Mouse Button String:Modifier Keys String:Path Type Integer:Pre- ascend String:Tree Textpath String:Operator	Tree (concrete)

Drop ( → page 80)	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 273)	e	String:Path Type Integer:Pre-ascend String:Tree Indexpath Integer:Delay before drop (milliseconds)	Tree (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 (concrete)
Expand Node by Indexpath ( → page 276)	e	String:Path Type Integer:Pre-ascend String:Indexpath	Tree (concrete)
Expand Node by Textpath ( → page 278)	e	String:Path Type Integer:Pre-ascend String:Textpath String:Operator	Tree (concrete)
Move ( → page 280)	e	String:Direction Integer:Number of Nodes Integer:Number of Clicks	Tree (concrete)

Select Context Menu Entry by Indexpath ( → page 82)	e	String:Indexpath Integer:Mouse Button	Graphics Component (abstract)	Component (abstract)
Select Context Menu Entry by Indexpath (Specify Position) ( → page 83)	e	Integer:x-position String:x-units Integer:y-position String:y-units String:Indexpath Integer:Mouse Button	Graphics Component (abstract)	Component (abstract)
Select Context Menu Entry by Textpath ( → page 86)	e	String:Textpath String:Operator Integer:Mouse Button	Graphics Component (abstract)	Component (abstract)
Select Context Menu Entry by Textpath (Specify Position) ( → page 88)	e	Integer:x-position String:x-units Integer:y-position String:y-units String:Menupath String:Operator Integer:Mouse Button	Graphics Component (abstract)	Component (abstract)
Select Node by Indexpath ( → page 282)	e	String:Path Type Integer:Pre-ascend String:Indexpath Integer:Number of Clicks Integer:Mouse Button String:Extend Selection	Tree (concrete)	

Select Node by Textpath ( → page 284)	e	String:Path Type Integer:Pre-ascend String:Textpath String:Operator Integer:Number of Clicks Integer:Mouse Button String:Extend Selection	Tree (concrete)
Show Text ( → page 91)	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 92)	e	Variable:Variable Name String:Property Name	Graphics Component (abstract)
Store Text at Mouse Position ( → page 287)	e	Variable:Variable Name	Tree (concrete)
Store Text of Selected Node ( → page 288)	e	Variable:Variable Name	Tree (concrete)
Wait for Component ( → page 93)	e	Integer:Timeout in ms Integer:Delay after Visibility	Graphics Component (abstract)

### 5.7.1.1 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.

### 5.7.1.2 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 15) .
- Select "*matches*" to use a regular expression ( → page 15) .
- Regardless of the operator used: If there is more than one match, the first 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.

### 5.7.1.3 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.

### 5.7.1.4 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.

## Components, Actions, and Parameters

- 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)**



### 5.7.1.5 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).

## Components, Actions, and Parameters

- 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 15) .
- Select "*matches*" to use a regular expression ( → page 15) .
- Regardless of the operator used: If there is more than one match, the first 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

Tree Table (swt)

## 5.7.2 Toolbar Item

### Description:

Toolbar items are buttons and their menus on a toolbar, such as the one in Jubula (Figure 5.33 → page 340 ). Each of the buttons on the toolbar is a toolbar item.



**Figure 5.33:** Toolbar items

Often, toolbar items have drop-down menus (Figure 5.34 → page 340 ).



**Figure 5.34:** Toolbar item menu

### Mapping toolbar items

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



**Figure 5.35:** Mapping toolbar items

### Synopsis:

- Toolbar Item (swt)
  - Button/Check Box/Radio Button (concrete)
    - \* Button Component (abstract)
      - Component with Text (abstract)
        - Graphics Component (abstract)

### New Actions

Name	Type	Parameters
Check Enablement of Entry by Indexpath ( → page 346)	e	String:Indexpath Boolean:Enabled

Check Enablement of Entry by Textpath ( → page 347)	e	String:Menupath String:Operator Boolean:Enabled
Check Existence of Entry by Indexpath ( → page 348)	e	String:Indexpath Boolean:Exists
Check Existence of Entry by Textpath ( → page 349)	e	String:Menupath String:Operator Boolean:Exists
Check Selection of Entry by Indexpath ( → page 350)	e	String:Indexpath Boolean:Selected
Check Selection of Entry by Textpath ( → page 351)	e	String:Menupath String:Operator Boolean:Selected
Select Menu Entry by Indexpath ( → page 352)	e	String:Indexpath
Select Menu Entry by Textpath ( → page 353)	e	String:Menupath String:Operator

## Inherited Actions

Name	Type	Parameters	Inherited from
Check Enablement ( → page 45)	c	Boolean:Enabled	Graphics Component (abstract)
Check Enablement of Context Menu Entry by Indexpath ( → page 46)	c	String:Indexpath Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)
Check Enablement of Context Menu Entry by Indexpath (Specify Position) ( → page 47)	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 50)	c	String:Textpath String:Operator Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)	Component (abstract)
Check Enablement of Context Menu Entry by Textpath (Specify Position) ( → page 51)	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)	Component (abstract)
Check Existence ( → page 54)	c	Boolean:Exists	Graphics Component (abstract)	Component (abstract)
Check Existence of Context Menu Entry by Indxpath ( → page 55)	c	String:Indxpath Boolean:Exists Integer:Mouse Button	Graphics Component (abstract)	Component (abstract)
Check Existence of Context Menu Entry by Indxpath (Specify Position) ( → page 56)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Indxpath Boolean:Exists Integer:Mouse Button	Graphics Component (abstract)	Component (abstract)
Check Existence of Context Menu Entry by Textpath ( → page 59)	c	String:Textpath String:Operator Boolean:Exists Integer:Mouse Button	Graphics Component (abstract)	Component (abstract)
Check Existence of Context Menu Entry by Textpath (Specify Position) ( → page 60)	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)	Component (abstract)

Check Focus ( → page 63)	c	Boolean:Has Focus	Graphics Component (abstract)	Component (abstract)
Check Property ( → page 64)	c	String:PropertyName String:PropertyValue String:Operator	Graphics Component (abstract)	Component (abstract)
Check Selection ( → page 27)	c	Boolean:Selected	Button Component (abstract)	Component (abstract)
Check Selection of Context Menu Entry by Indexpath ( → page 65)	c	String:Indexpath Boolean:Selected Integer:Mouse Button	Graphics Component (abstract)	Component (abstract)
Check Selection of Context Menu Entry by Indexpath (Specify Position) ( → page 67)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Indexpath Boolean:Selected Integer:Mouse Button	Graphics Component (abstract)	Component (abstract)
Check Selection of Context Menu Entry by Textpath ( → page 69)	c	String:Textpath String:Operator Boolean:Selected Integer:Mouse Button	Graphics Component (abstract)	Component (abstract)
Check Selection of Context Menu Entry by Textpath (Specify Position) ( → page 71)	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)	Component (abstract)

Check Text ( → page 32)	c	String:Text String:Operator	Component with Text (ab- stract)	
Click ( → page 74)	e	Integer:Number of Clicks Integer:Mouse Button	Graphics Com- ponent (ab- stract)	
Click in Component ( → page 75)	e	Integer:Number of Clicks Integer:Mouse Button Integer:x- position String:x-units Integer:y- position String:y-units	Graphics Com- ponent (ab- stract)	
Drag ( → page 77)	e	Integer:Mouse Button String:Modifier Keys Integer:x- position String:x-units Integer:y- position String:y-units	Graphics Com- ponent (ab- stract)	
Drop ( → page 80)	e	Integer:x- position String:x-units Integer:y- position String:y-units Integer:Delay before drop (milliseconds)	Graphics Com- ponent (ab- stract)	

Select Context Menu Entry by Indxpath ( → page 82)	e	String:Indxpath Integer:Mouse Button	Graphics Component (abstract)	Component (abstract)
Select Context Menu Entry by Indxpath (Specify Position) ( → page 83)	e	Integer:x-position String:x-units Integer:y-position String:y-units String:Indxpath Integer:Mouse Button	Graphics Component (abstract)	Component (abstract)
Select Context Menu Entry by Textpath ( → page 86)	e	String:Textpath String:Operator Integer:Mouse Button	Graphics Component (abstract)	Component (abstract)
Select Context Menu Entry by Textpath (Specify Position) ( → page 88)	e	Integer:x-position String:x-units Integer:y-position String:y-units String:Menupath String:Operator Integer:Mouse Button	Graphics Component (abstract)	Component (abstract)
Show Text ( → page 91)	e	String:Text Integer:Text Size (in points) Integer:Time per Word (in milliseconds) Integer:Window Width (in pixels)	Graphics Component (abstract)	Component (abstract)
Store Property ( → page 92)	e	Variable:Variable Name String:Property Name	Graphics Component (abstract)	Component (abstract)
Store Value ( → page 33)	e	Variable:Variable Name	Component with Text (abstract)	Component (abstract)

Wait for Component ( → page 93)	e	Integer:Timeout in ms Integer:Delay after Visibility	Graphics Component (abstract)
------------------------------------	---	---	----------------------------------

### 5.7.2.1 Check Enablement of Entry by Indexpath

(Toolbar 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.

## 5.7.2.2 Check Enablement of Entry by Textpath

(Toolbar 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 15) .
- Select "*matches*" to use a regular expression ( → page 15) .
- Regardless of the operator used: If there is more than one match, the first 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.

### 5.7.2.3 Check Existence of Entry by Indexpath

(Toolbar 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.

## 5.7.2.4 Check Existence of Entry by Textpath

(Toolbar 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 15) .
- Select "*matches*" to use a regular expression ( → page 15) .
- Regardless of the operator used: If there is more than one match, the first 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.

### 5.7.2.5 Check Selection of Entry by Indexpath

(Toolbar 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.

## 5.7.2.6 Check Selection of Entry by Textpath

(Toolbar 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 15) .
- Select "*matches*" to use a regular expression ( → page 15) .

- Regardless of the operator used: If there is more than one match, the first 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.

### 5.7.2.7 Select Menu Entry by Indexpath

(Toolbar 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)**

---

## 5.7.2.8 Select Menu Entry by Textpath

(Toolbar 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 15) .
- Select "*matches*" to use a regular expression ( → page 15) .
- Regardless of the operator used: If there is more than one match, the first found will be chosen.

### 5.7.3 Tree Table

#### Description:

A tree table is an SWT component which has some of the properties 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 379) for more details.

#### Synopsis:

- Tree Table (swt)
  - (SWT) Tree (swt)
    - \* Tree (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 45)	c	Boolean:Enabled	Graphics Component (abstract)
Check Enablement of Context Menu Entry by Indxpath ( → page 46)	c	String:Indxpath Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)
Check Enablement of Context Menu Entry by Indxpath (Specify Position) ( → page 47)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Indxpath Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)
Check Enablement of Context Menu Entry by Textpath ( → page 50)	c	String:Textpath String:Operator Boolean:Enabled Integer:Mouse Button	Graphics Component (abstract)
Check Enablement of Context Menu Entry by Textpath (Specify Position) ( → page 51)	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 54)	c	Boolean:Exists	Graphics Component (abstract)	Component (abstract)
Check Existence of Context Menu Entry by Indexpath ( → page 55)	c	String:Indexpath Boolean:Exists Integer:Mouse Button	Graphics Component (abstract)	Component (abstract)
Check Existence of Context Menu Entry by Indexpath (Specify Position) ( → page 56)	c	Integer:x- position String:x-units Integer:y- position String:y-units String:Indexpath Boolean:Exists Integer:Mouse Button	Graphics Component (abstract)	Component (abstract)
Check Existence of Context Menu Entry by Textpath ( → page 59)	c	String:Textpath String:Operator Boolean:Exists Integer:Mouse Button	Graphics Component (abstract)	Component (abstract)
Check Existence of Context Menu Entry by Textpath (Specify Position) ( → page 60)	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)	Component (abstract)
Check Existence of Node by Indexpath ( → page 258)	c	String:Path Type Integer:Pre- ascend String:Indexpath Boolean:Exists	Tree (concrete)	
Check Existence of Node by Textpath ( → page 259)	c	String:Path Type Integer:Pre- ascend String:Textpath String:Operator Boolean:Exists	Tree (concrete)	

Check Focus ( → page 63)	c	Boolean:Has Focus	Graphics Component (abstract)	Component (abstract)
Check Property ( → page 64)	c	String:PropertyName String:PropertyValue String:Operator	Graphics Component (abstract)	Component (abstract)
Check Selection of Checkbox by Indexpath ( → page 332)	c	String:Path Type Integer:Pre-ascend String:Indexpath Boolean:Checked	(SWT) Tree (swt)	
Check Selection of Checkbox by Textpath ( → page 334)	c	String:Path Type Integer:Pre-ascend String:Textpath String:Operator Boolean:Checked	(SWT) Tree (swt)	
Check Selection of Checkbox on Selected Node ( → page 336)	c	Boolean:Checked	(SWT) Tree (swt)	
Check Selection of Context Menu Entry by Indexpath ( → page 65)	c	String:Indexpath Boolean:Selected Integer:Mouse Button	Graphics Component (abstract)	Component (abstract)
Check Selection of Context Menu Entry by Indexpath (Specify Position) ( → page 67)	c	Integer:x-position String:x-units Integer:y-position String:y-units String:Indexpath Boolean:Selected Integer:Mouse Button	Graphics Component (abstract)	Component (abstract)

Check Selection of Context Menu Entry by Textpath ( → page 69)	c	String:Textpath String:Operator Boolean:Selected Integer:Mouse Button	Graphics Component (abstract)
Check Selection of Context Menu Entry by Textpath (Specify Position) ( → page 71)	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 261)	c	String:Text String:Operator	Tree (concrete)
Check Text of Selected Node(s) ( → page 262)	c	String:Text String:Operator	Tree (concrete)
Click ( → page 74)	e	Integer:Number of Clicks Integer:Mouse Button	Graphics Component (abstract)
Click in Component ( → page 75)	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 Ind- expath ( → page 263)	e	String:Path Type Integer:Pre- ascend String:Indexpath	Tree (concrete)
Collapse Node by Textpath ( → page 265)	e	String:Path Type Integer:Pre- ascend String:Textpath String:Operator	Tree (concrete)
Drag ( → page 77)	e	Integer:Mouse Button String:Modifier Keys Integer:x- position String:x-units Integer:y- position String:y-units	Graphics Com- ponent (ab- stract)
Drag Node by Index- path ( → page 267)	e	Integer:Mouse Button String:Modifier Keys String:Path Type Integer:Pre- ascend String:Tree Ind- expath	Tree (concrete)
Drag Node by Textpath ( → page 270)	e	Integer:Mouse Button String:Modifier Keys String:Path Type Integer:Pre- ascend String:Tree Textpath String:Operator	Tree (concrete)

Drop ( → page 80)	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 273)	e	String:Path Type Integer:Pre-ascend String:Tree Indexpath Integer:Delay before drop (milliseconds)	Tree (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 (concrete)
Expand Node by Indexpath ( → page 276)	e	String:Path Type Integer:Pre-ascend String:Indexpath	Tree (concrete)
Expand Node by Textpath ( → page 278)	e	String:Path Type Integer:Pre-ascend String:Textpath String:Operator	Tree (concrete)
Move ( → page 280)	e	String:Direction Integer:Number of Nodes Integer:Number of Clicks	Tree (concrete)

Select Context Menu Entry by Indexpath ( → page 82)	e	String:Indexpath Integer:Mouse Button	Graphics Component (abstract)	Component (abstract)
Select Context Menu Entry by Indexpath (Specify Position) ( → page 83)	e	Integer:x-position String:x-units Integer:y-position String:y-units String:Indexpath Integer:Mouse Button	Graphics Component (abstract)	Component (abstract)
Select Context Menu Entry by Textpath ( → page 86)	e	String:Textpath String:Operator Integer:Mouse Button	Graphics Component (abstract)	Component (abstract)
Select Context Menu Entry by Textpath (Specify Position) ( → page 88)	e	Integer:x-position String:x-units Integer:y-position String:y-units String:Menupath String:Operator Integer:Mouse Button	Graphics Component (abstract)	Component (abstract)
Select Node by Indexpath ( → page 282)	e	String:Path Type Integer:Pre-ascend String:Indexpath Integer:Number of Clicks Integer:Mouse Button String:Extend Selection	Tree (concrete)	

Select Node by Textpath ( → page 284)	e	String:Path Type Integer:Pre-ascend String:Textpath String:Operator Integer:Number of Clicks Integer:Mouse Button String:Extend Selection	Tree (concrete)
Show Text ( → page 91)	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 92)	e	Variable:Variable Name String:Property Name	Graphics Component (abstract)
Store Text at Mouse Position ( → page 287)	e	Variable:Variable Name	Tree (concrete)
Store Text of Selected Node ( → page 288)	e	Variable:Variable Name	Tree (concrete)
Toggle Checkbox on Node by Indexpath ( → page 336)	e	String:Path Type Integer:Pre-ascend String:Indexpath	(SWT) Tree (swt)
Toggle Checkbox on Node by Textpath ( → page 337)	e	String:Path Type Integer:Pre-ascend String:Textpath String:Operator	(SWT) Tree (swt)
Wait for Component ( → page 93)	e	Integer:Timeout in ms Integer:Delay after Visibility	Graphics Component (abstract)

## 5.7.3.1 Check Selection by Value(s) (Specify Column)

(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 15) .
- Select "*matches*" to use a regular expression ( → page 15) .
- Regardless of the operator used: If there is more than one match, the first 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*.

### 5.7.3.2 Select Entry by Indexpath (Specify Column)

(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.
- Jubula realises this by:
  - Searching for the node and column you specify.
  - If the tree is not expanded, Jubula expands it.
  - 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*.

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.**

---

### 5.7.3.3 Select Entry by Textpath (Specify Column)

(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.
- Jubula realises this by:
  - Searching for the node and column you specify.
  - If the tree is not expanded, Jubula expands it.
  - 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 15) .

- Select "*matches*" to use a regular expression ( → page 15) .
- Regardless of the operator used: If there is more than one match, the first 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*.

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.**

## 5.8 Deprecated Actions

The following actions have been deprecated for this release, and will be removed completely from Jubula in future releases. Each deprecated action is named below, with suggested alternatives/workarounds given.

---

**action:** abstract - Graphics Component - Select Context Menu Entry by Indexpath (DEPRECATED)

**alternative:** Use the new action of the same name.

---

**action:** abstract - Graphics Component - Select Context Menu Entry by Indexpath (Specify Position) (DEPRECATED)

**alternative:** Use the new action of the same name.

---

**action:** abstract - Graphics Component - Select Context Menu Entry by Textpath (DEPRECATED)

**alternative:** Use the new action of the same name.

---

**action:** abstract - Graphics Component - Select Context Menu Entry by Textpath (Specify Position) (DEPRECATED)

**alternative:** Use the new action of the same name.

---

**action:** concrete - Application - Replace Text (DEPRECATED)

**alternative:** Component with Text Input - Replace Text

---

**action:** concrete - Table - Check Editability (Specify Cell by Index) (DEPRECATED)

**alternative:** Table - Check Editability (Specify Cell)

---

**action:** concrete - Table - Check Text (Specify Cell by Index) (DEPRECATED)

**alternative:** Table - Check Text (Specify Cell)

---

**action:** concrete - Table - Drag Cell by Index (DEPRECATED)

**alternative:** Table - Drag Cell

---

**action:** concrete - Table - Drag Cell from Column by Index (DEPRECATED)

**alternative:** Table - Drag Cell from Column

---

**action:** concrete - Table - Drag Cell from Row by Index (DEPRECATED)

**alternative:** Table - Drag Cell from Row

---

**action:** concrete - Table - Drop on Cell by Index (DEPRECATED)

**alternative:** Table - Drop Cell

---

**action:** concrete - Table - Drop on Cell from Column by Index (DEPRECATED)

**alternative:** Table - Drop on Cell from Column

---

**action:** concrete - Table - Drop on Cell from Row by Index (DEPRECATED)

**alternative:** Table - Drop on Cell from Row

---

**action:** concrete - Table - Input Text (Specify Cell by Index) (DEPRECATED)

**alternative:** Table - Input Text (Specify Cell)

---

**action:** concrete - Table - Replace Text (Specify Cell by Index) (DEPRECATED)

**alternative:** Table - Replace Text (Specify Cell)

---

**action:** concrete - Table - Select Cell by Index (DEPRECATED)

**alternative:** Table - Select Cell

---

**action:** concrete - Table - Select Context Menu Entry by Ind-  
expath (Selected Cell) (DEPRECATED)

**alternative:** Graphics Component - Context Menu by Index-  
path

---

**action:** concrete - Table - Select Context Menu Entry by Ind-  
expath (Specify Cell) (DEPRECATED)

**alternative:** Graphics Component - Context Menu by Index-  
path (Specify Position)

---

**action:** concrete - Table - Select Context Menu Entry by  
Textpath (Selected Cell) (DEPRECATED)

**alternative:** Graphics Component - Context Menu by  
Textpath

---

**action:** concrete - Table - Select Context Menu Entry by  
Textpath (Specify Cell) (DEPRECATED)

**alternative:** Graphics Component - Context Menu by  
Textpath (Specify Position)

---

**action:** concrete - Table - Select Row (DEPRECATED)

**alternative:** Table - Select Row

---

**action:** concrete - Table - Select Value from Column (DEPRE-  
CATED)

**alternative:** Table - Select Value from Column

---

**action:** concrete - Table - Select Value from Column by Index  
(DEPRECATED)

**alternative:** Table - Select Value from Column

---

**action:** concrete - Table - Select Value from Row by Index  
(DEPRECATED)

**alternative:** Table - Select Value from Row

---

**action:** concrete - Table - Store Value (Specify Cell by Index)  
(DEPRECATED)

**alternative:** Table - Store Value (Specify Cell)

---

**action:** concrete - Tree - Select Context Menu Entry by Indexpath (Node specified by indexpath) (DEPRECATED)

**alternative:** Graphics Component - Context Menu by Indexpath (Specify Position)

---

**action:** concrete - Tree - Select Context Menu Entry by Indexpath (Node specified by textpath) (DEPRECATED)

**alternative:** Graphics Component - Context Menu by Textpath (Specify Position)

---

**action:** concrete - Tree - Select Context Menu Entry by Indexpath (Selected Node) (DEPRECATED)

**alternative:** Graphics Component - Context Menu by Textpath

---

**action:** concrete - Tree - Select Context Menu Entry by Textpath (Node specified by indexpath) (DEPRECATED)

**alternative:** Graphics Component - Context Menu by Indexpath (Specify Position)

---

**action:** concrete - Tree - Select Context Menu Entry by Textpath (Selected Node) (DEPRECATED)

**alternative:** Graphics Component - Context Menu by Textpath

---

**action:** concrete - Tree - Select Context Menu Entry by Textpath (Node specified by textpath) (DEPRECATED)

**alternative:** Graphics Component - Context Menu by Textpath (Specify Position)

---

**action:** html - Browser - Select window

**alternative:** This action has not been replaced in the current version.

---

**action:** html - HTML Hyperlink - Follow Link

**alternative:** Use the action *Click on Graphics Component* instead of this action.

## Chapter 6

# Constants for lists, combo boxes, trees and menus

Many of the Jubula actions use parameters for which you need to enter a "path" (e.g. menupaths, a path through a tree etc.) or a "list" (e.g. to select multiple items from a list or combo box).

The following are the characters you must use to enter lists and paths:

**Path Separator** Hierarchical components (e.g. menus, trees) use slash ' / ' as a value separator. An example menupath is therefore:

`File/Open`

An example indexpath could be:

`1/2`

**Value Separator** For components which allow more than one item to be selected or checked at once, use a comma ' , ' to separate items. For example, to select the values "blue", "green" and "orange" from a list, the parameter would look like this:

`blue,green,orange`

---

**Do not enter a space between list items.**

---

If you are testing a component whose actions use these separators, and you need to enter a comma or slash as a part of the actual parameter, you will need to mask the comma or slash ( → page 379) .



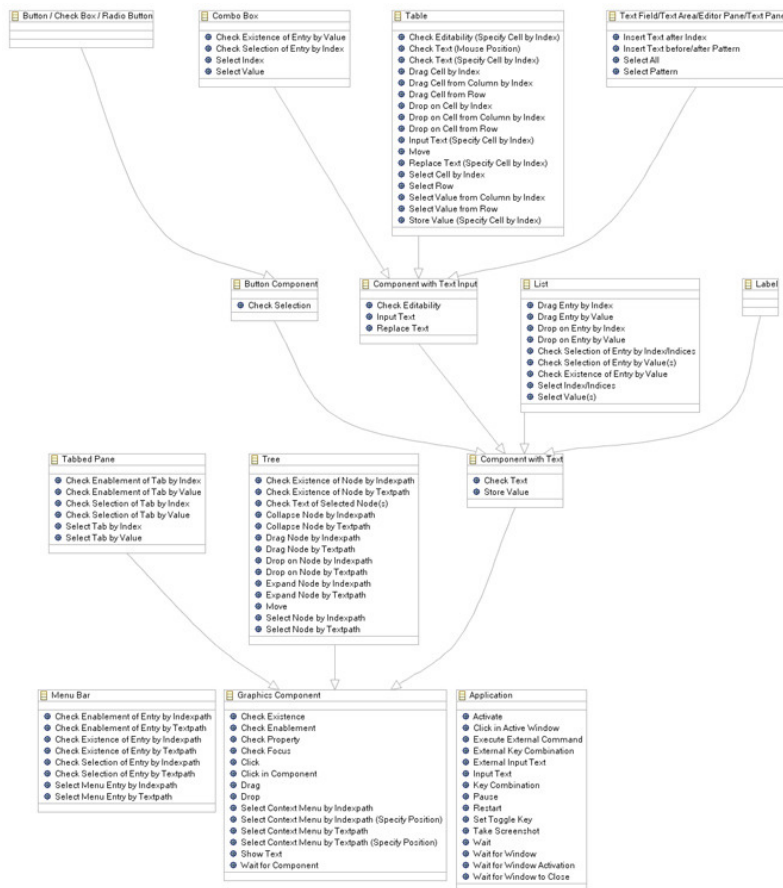
## **6.1 Parameters which use Indices**

All parameters which use indices (e.g. selecting from a list, or entering a menupath by the index) start at index ' 1 '. For example, the first row in a table is indexed with ' 1 '.

# Chapter 7

## Overview of Components

The following diagram shows which abstract components Jubula uses and for which actions they can be used for.



**Figure 7.1:** Jubula component structure

Figure 7.1 provides an overview of Jubula's 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 8

## Using Relative Paths

- In the AUT configuration and in the Properties View you can enter paths to files Jubula needs.
- For many paths, you do not have to enter the whole path. Instead, you can use a relative path.
- The root for relative paths depends on which path you are entering:

### **Paths to .JAR files, JRE binaries, classpaths and class-names**

These items reside on the machine on which the AUT Agent is installed.

- The paths to these files are entered in the AUT configuration.
- They are relative to the AUT *working directory*, if you have entered one.
- If no AUT *working directory* is given, these paths are relative to the "server" directory.

### **Paths to Excel files**

External data files reside on the machine on which the ITE is installed.

- When you use an Excel file as an external data source, you must enter the path to the Excel file in the Properties View.
- The default root for external data sources is your workspace directory.
- You can also enter a different root in the preference pages.

### **Paths to screenshots taken with the *take screenshot* actions**

- When you take a screenshot in your test, you must specify a place to save the screenshot.
- Paths you enter are relative to the AUT *base directory*, if one has been specified. Otherwise, they are relative to the "server" directory.
- Screenshots taken during the test can only be saved on the machine on which the AUT Agent is running.

### **Paths to commands executed with the *execute external command* action**

- When you use the action *execute external command*, you specify whether the command should be executed locally or remotely.
- If you execute a local command, the relative path to it uses the test data directory as specified in the preferences as its root, if you have specified one. If not, the root is your workspace directory.
- If you execute a remote command, the root directory for the relative path is the AUT *working directory*, if you have defined one. If not, the default root directory is the "server" directory.

## Chapter 9

# Special characters in Jubula

Jubula has a few symbols with a special function (Figure 9.2 → page 380 ). If you want to use these symbols as a part of a text entry or verification, you must mask them with a backslash.

### 9.1 Verbatim text symbol

If you have a lot of characters that need to be masked, you can place the data in single quote marks:

'DATA'

This means that any special functions of the characters between the quote marks are ignored.

### 9.2 General special characters

Name	Symbol	Function	Masked version
Open curly brackets	{	Delimits reference/variable names within a concatenated parameter value	\{
Close curly brackets	}	Delimits reference/variable names within a concatenated parameter value	\}
Single quote	'	Delimits a verbatim area in a parameter	\'
Two single quotes	"	Denotes an empty string	\' \'
Dollar	\$	Denotes a variable	\\$
Equals	=	Denotes a referenced parameter	\=
Question mark	?	Denotes a function	?
Backslash	\	The escape symbol	\\

**Figure 9.1:** Special symbols in Jubula

## 9.3 Symbols with special meanings for certain parameters

In addition to these general symbols, there are a few symbols which have a special meaning for particular components (Figure 9.3 → page 381 ). They must be masked if you want to use them without their special function *for these components only*. Masking these symbols must be done with a double backslash.

Name	Symbol	Function	Masked version
Comma	,	Separator for lists	\\,
Forward slash	/	Separator for paths in menus and trees	\\/

**Figure 9.2:** Symbols for components



# Chapter 10

## Language Codes

Locale	Country	Code
Albanian (Albania)	Albania	sq_AL
Arabic (Algeria)	Algeria	ar_DZ
Arabic (Bahrain)	Bahrain	ar_BH
Arabic (Egypt)	Egypt	ar_EG
Arabic (Iraq)	Iraq	ar_IQ
Arabic (Jordan)	Jordan	ar_JO
Arabic (Kuwait)	Kuwait	ar_KW
Arabic (Lebanon)	Lebanon	ar_LB
Arabic (Libya)	Libya	ar_LY
Arabic (Morocco)	Morocco	ar_MA
Arabic (Oman)	Oman	ar_OM
Arabic (Qatar)	Qatar	ar_QA
Arabic (Saudi Arabia)	Saudi Arabia	ar_SA
Arabic (Sudan)	Sudan	ar_SD
Arabic (Syria)	Syria	ar_SY
Arabic (Tunisia)	Tunisia	ar_TN
Arabic (United Arab Emirates)	United Arab Emirates	ar_AE
Arabic (Yemen)	Yemen	ar_YE
Belarusian (Belarus)	Belarus	be_BY
Bulgarian (Bulgaria)	Bulgaria	bg_BG
Catalan (Spain)	Spain	ca_ES
Chinese (China)	China	zh_CN
Chinese (Hong Kong)	Hong Kong	zh_HK
Chinese (Taiwan)	Taiwan	zh_TW
Croatian (Croatia)	Croatia	hr_HR
Czech (Czech Republic)	Czech Republic	cs_CZ
Danish (Denmark)	Denmark	da_DK

<b>Locale</b>	<b>Country</b>	<b>Code</b>
Dutch (Belgium)	Belgium	nl_BE
Dutch (Netherlands)	Netherlands	nl_NL
English (Australia)	Australia	en_AU
English (Canada)	Canada	en_CA
English (India)	India	en_IN
English (Ireland)	Ireland	en_IE
English (New Zealand)	New Zealand	en_NZ
English (South Africa)	South Africa	en_ZA
English (United Kingdom)	United Kingdom	en_GB
English (United States)	United States	en_US
Estonian (Estonia)	Estonia	et_EE
Finnish (Finland)	Finland	fi_FI
French (Belgium)	Belgium	fr_BE
French (Canada)	Canada	fr_CA
French (France)	France	fr_FR
French (Luxembourg)	Luxembourg	fr_LU
French (Switzerland)	Switzerland	fr_CH
German (Austria)	Austria	de_AT
German (Germany)	Germany	de_DE
German (Luxembourg)	Luxembourg	de_LU
German (Switzerland)	Switzerland	de_CH
Greek (Greece)	Greece	el_GR
Hebrew (Israel)	Israel	iw_IL
Hindi (India)	India	hi_IN
Hungarian (Hungary)	Hungary	hu_HU
Icelandic (Iceland)	Iceland	is_IS
Italian (Italy)	Italy	it_IT
Italian (Switzerland)	Switzerland	it_CH
Japanese (Japan)	Japan	ja_JP
Korean (South Korea)	South Korea	ko_KR
Latvian (Latvia)	Latvia	lv_LV
Lithuanian (Lithuania)	Lithuania	lt_LT
Macedonian (Macedonia)	Macedonia	mk_MK
Norwegian (Norway)	Norway	no_NO
Norwegian (Norway, Nynorsk)	Norway	no_NO_NY
Polish (Poland)	Poland	pl_PL
Portuguese (Brazil)	Brazil	pt_BR
Portuguese (Portugal)	Portugal	pt_PT

Locale	Country	Code
Romanian (Romania)	Romania	ro_RO
Russian (Russia)	Russia	ru_RU
Slovak (Slovakia)	Slovakia	sk_SK
Slovenian (Slovenia)	Slovenia	sl_SI
Spanish (Argentina)	Argentina	es_AR
Spanish (Bolivia)	Bolivia	es_BO
Spanish (Chile)	Chile	es_CL
Spanish (Colombia)	Colombia	es_CO
Spanish (Costa Rica)	Costa Rica	es_CR
Spanish (Dominican Republic)	Dominican Republic	es_DO
Spanish (Ecuador)	Ecuador	es_EC
Spanish (El Salvador)	El Salvador	es_SV
Spanish (Guatemala)	Guatemala	es_GT
Spanish (Honduras)	Honduras	es_HN
Spanish (Mexico)	Mexico	es_MX
Spanish (Nicaragua)	Nicaragua	es_NI
Spanish (Panama)	Panama	es_PA
Spanish (Paraguay)	Paraguay	es_PY
Spanish (Peru)	Peru	es_PE
Spanish (Puerto Rico)	Puerto Rico	es_PR
Spanish (Spain)	Spain	es_ES
Spanish (Uruguay)	Uruguay	es_UY
Spanish (Venezuela)	Venezuela	es_VE
Swedish (Sweden)	Sweden	sv_SE
Thai (Thailand)	Thailand	th_TH
Thai (Thailand,TH)	Thailand	th_TH_TH
Turkish (Turkey)	Turkey	tr_TR
Ukrainian (Ukraine)	Ukraine	uk_UA
Vietnamese (Vietnam)	Vietnam	vi_VN



# Chapter 11

## Keyboard layout files

For SWT and RCP AUT's, you need to define a keyboard layouts for the AUT (i.e. which keyboard layout is set for the system on which the AUT runs).

Jubula already has the keyboard layouts for German (Germany) and English (US) saved. If you want to use another keyboard layout, follow the instructions in the next section to create one.

### 11.1 Creating a keyboard layout file

1. The name of the file must be in the following format:  
keyboardmapping\_<language>\_<COUNTRY>.properties  
e.g. for US English: keyboardmapping\_en\_US.properties  
See the section in this document ( → page 383) for language codes.
2. In the file, enter the codes for all characters which require you to press a modifier key.
3. The format for these codes is:  
<Character>=<Modifier>+<Character without modifier>  
e.g. for the @ character: @=shift+2
4. The following symbols must be escaped with a backslash:  
! = : \ ,  
e.g. \!=shift+1 for !
5. There are pictures of various keyboards on the following website:  
<http://www.uni-regensburg.de/EDV/Misc/KeyBoards/>

6. To add the keyboard layout file to Jubula, you must create a fragment containing your newly added keyboard layout file(s). The host plugin for the fragment is:

*org.eclipse.jubula.client.core*

The path within the fragment must be:

*resources/keyboard\_mapping/<name>.properties*



---

**If you name the file with the locale code, then Jubula will display the keyboard layout in plain text in the AUT configuration dialog.**

---

# Chapter 12

## Debugging

### 12.1 Remote Debugging

You can remote debug an AUT which has been started by Jubula by simply modifying or creating a AUT configuration. The following options must be added to your JRE Arguments:

```
-Xdebug  
-Xnoagent  
-Xrunjdwpt:transport=dt_socket,server=y,suspend=y,address=4711  
-Djava.compiler=NONE
```

- The "suspend=y/n" option determines whether the JVM waits with the execution of the program until the debugger has connected itself or not.
- The "address=<portNo>" determines the port the debugger has to connect to.

---

**Not all JREs support the arguments listed above. If you are not using a Java Standard Edition JRE to start your AUT, you will likely not be able to use this method to remotely debug your AUT.**

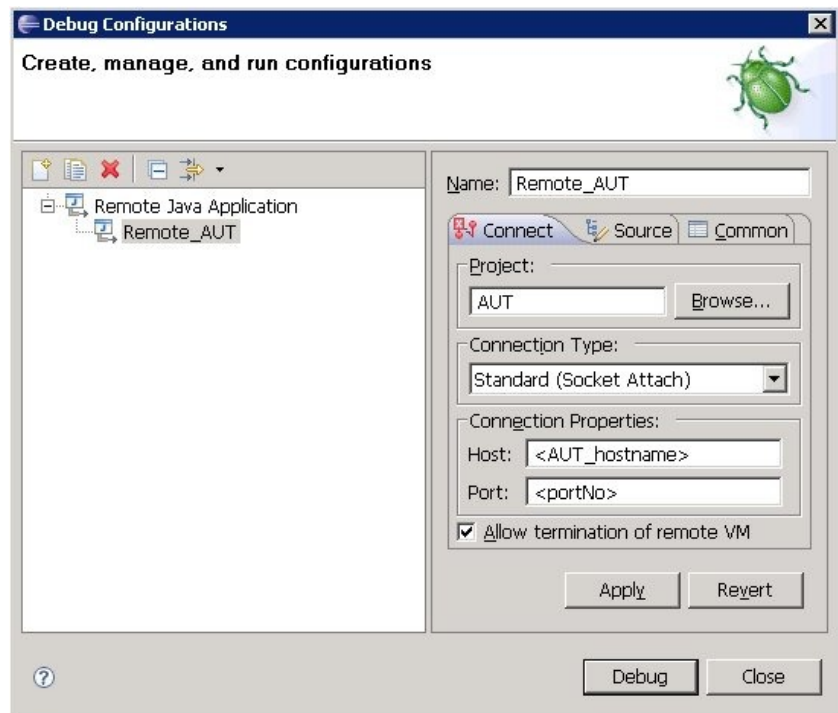
---



#### 12.1.1 Configuring Eclipse for remote debugging

Create a new "Remote Java Application" in the "Launch configuration" dialog and enter the hostname for the computer on which the AUT Agent is running. The specified port has

to be the same which you used in the AUT configuration (e.g. 4711).



**Figure 12.1:** Launching eclipse remote debug application

To get your AUT running using remote debugging options you have to take the following steps:

1. Start the AUT Agent
2. Connect the ITE with the AUT Agent, load the Project in Jubula and invoke the startup of the AUT in the ITE.
3. If "suspend=y", you now have to run your "Remote Java Application"-configuration in Eclipse, as the JVM is waiting for the debugger to connect before starting the application. As soon as you are connected you should see the default debug behaviour in your debug perspective of Eclipse.
4. Your AUT has now been launched with the ability to use remote debugging.

If you wish to debug an RCP AUT which is launched with an executable file, you must pass these JRE arguments via the AUT arguments and "-vmargs". When using this option, make sure you overwrite all JRE options that are defined in the "config.ini" of your RCP launcher.

# Chapter 13

## Index

### (SWT) Tree, 325

- Check Selection of Checkbox by Indexpath, 332
- Check Selection of Checkbox by Textpath, 334
- Check Selection of Checkbox on Selected Node, 336
- Toggle Checkbox on Node by Indexpath, 336
- Toggle Checkbox on Node by Textpath, 337

### Actions

- Activate, 96
- Check Editability, 39
- Check Editability (Specify Cell), 195
- Check Editability of Cell (Mouse Position), 197
- Check Editability of Selected Cell, 197
- Check Enablement, 45
- Check Enablement of Context Menu Entry by Indexpath, 46
- Check Enablement of Context Menu Entry by Indexpath (Specify Position), 47
- Check Enablement of Context Menu Entry by Textpath, 50
- Check Enablement of Context Menu Entry by Textpath (Specify Position), 51
- Check Enablement of Entry by Indexpath, 165, 346
- Check Enablement of Entry by Textpath, 166, 347
- Check Enablement of Tab by Index, 179
- Check Enablement of Tab by Value, 180
- Check Existence, 54
- Check Existence of Context Menu Entry by Indexpath, 55
- Check Existence of Context Menu Entry by Indexpath (Specify Position), 56
- Check Existence of Context Menu Entry by Textpath, 59
- Check Existence of Context Menu Entry by Textpath (Specify Position), 60

Check Existence of Entry by Indexpath, 167, 348  
Check Existence of Entry by Textpath, 168, 349  
Check Existence of Entry by Value, 135, 151  
Check Existence of Node by Indexpath, 258  
Check Existence of Node by Textpath, 259  
Check Existence of Tab, 181  
Check Existence of Value in Column, 198  
Check Existence of Value in Row, 200  
Check Existence of Window, 97  
Check Figure Existence, 297  
Check Figure Property, 298  
Check Focus, 63  
Check Numeric Values, 98  
Check Property, 64  
Check Selection, 27  
Check Selection by Value(s) (Specify Column), 363  
Check Selection of Checkbox by Indexpath, 332  
Check Selection of Checkbox by Textpath, 334  
Check Selection of Checkbox on Selected Node, 336  
Check Selection of Context Menu Entry by Indexpath, 65  
Check Selection of Context Menu Entry by Indexpath  
(Specify Position), 67  
Check Selection of Context Menu Entry by Textpath, 69  
Check Selection of Context Menu Entry by Textpath (Spec-  
ify Position), 71  
Check Selection of Entry by Index, 136  
Check Selection of Entry by Index/Indices, 152  
Check Selection of Entry by Indexpath, 169, 350  
Check Selection of Entry by Textpath, 169, 351  
Check Selection of Entry by Value(s), 153  
Check Selection of Tab by Index, 182  
Check Selection of Tab by Value, 182  
Check String Values, 99  
Check Text, 32  
Check Text (Mouse Position), 201, 261  
Check Text (Specify Cell), 202  
Check Text of Selected Node(s), 262  
Check Text of Tab by Index, 183  
Check Tool Existence, 300  
Check URL, 323  
Click, 74  
Click Connection, 301  
Click Figure, 304  
Click in Active Window, 100  
Click in Component, 75

- Click in Figure, 305
- Collapse Node by Indexpath, 263
- Collapse Node by Textpath, 265
- Copy Text to Clipboard, 102
- Drag, 77
- Drag Cell, 204
- Drag Cell from Column, 209
- Drag Cell from Row, 212
- Drag Entry by Index, 154
- Drag Entry by Value, 155
- Drag Figure, 308
- Drag Node by Indexpath, 267
- Drag Node by Textpath, 270
- Drop, 80
- Drop on Cell, 215
- Drop on Cell from Column, 218
- Drop on Cell from Row, 220
- Drop on Entry by Index, 158
- Drop on Entry by Value, 158
- Drop on Figure, 311
- Drop on Node by Indexpath, 273
- Drop on Node by Textpath, 274
- Execute External Command, 103
- Expand Node by Indexpath, 276
- Expand Node by Textpath, 278
- External Input Text, 104
- External Key Combination, 105
- Go Back, 316
- Input Text, 39, 107
- Input Text (Specify Cell), 222
- Insert Text after Index, 247
- Insert Text before/after Pattern, 248
- Key Combination, 108
- Manual Test Step, 110
- Move, 224, 280
- Open URL, 316
- Pause Test Execution, 110
- Read Timer, 111
- Reload page, 317
- Replace Text, 40
- Replace Text (Specify Cell), 227
- Restart, 111
- Select All, 249
- Select Cell, 229
- Select Context Menu Entry by Indexpath, 82

Select Context Menu Entry by Indexpath (Specify Position), 83  
Select Context Menu Entry by Textpath, 86  
Select Context Menu Entry by Textpath (Specify Position), 88  
Select Entry by Index, 137  
Select Entry by Index/Indices, 160  
Select Entry by Indexpath (Specify Column), 364  
Select Entry by Textpath (Specify Column), 366  
Select Entry by Value, 137  
Select Entry by Value(s), 161  
Select Menu Entry by Indexpath, 171, 352  
Select Menu Entry by Textpath, 171, 353  
Select Node by Indexpath, 282  
Select Node by Textpath, 284  
Select Pattern, 249  
Select Tab by Index, 184  
Select Tab by Value, 185  
Select Tool, 314  
Select Value from Column, 233  
Select Value from Row, 236  
Set Toggle Key, 112  
Show Text, 91  
Start Timer, 113  
Store Property, 92  
Store Text at Mouse Position, 287  
Store Text of Selected Node, 288  
Store Value, 33, 114  
Store Value (Specify Cell), 238  
Store Value at Mouse Position, 240  
Take Screenshot, 114  
Take Screenshot of Active Window, 116  
Toggle Checkbox on Node by Indexpath, 336  
Toggle Checkbox on Node by Textpath, 337  
Wait, 118  
Wait for Component, 93, 172  
Wait for Window, 118  
Wait for Window Activation, 120  
Wait for Window to Close, 121  
Application, 94  
    Activate, 96  
    Check Existence of Window, 97  
    Check Numeric Values, 98  
    Check String Values, 99  
    Click in Active Window, 100

- Copy Text to Clipboard, 102
- Execute External Command, 103
- External Input Text, 104
- External Key Combination, 105
- Input Text, 107
- Key Combination, 108
- Manual Test Step, 110
- Pause Test Execution, 110
- Read Timer, 111
- Restart, 111
- Set Toggle Key, 112
- Start Timer, 113
- Store Value, 114
- Take Screenshot, 114
- Take Screenshot of Active Window, 116
- Wait, 118
- Wait for Window, 118
- Wait for Window Activation, 120
- Wait for Window to Close, 121
- Browser, 316
  - Go Back, 316
  - Open URL, 316
  - Reload page, 317
- Button Component, 22
  - Check Selection, 27
- Button/Check Box/Radio Button, 123
- Combo Box, 129
  - Check Existence of Entry by Value, 135
  - Check Selection of Entry by Index, 136
  - Select Entry by Index, 137
  - Select Entry by Value, 137
- Component
  - (SWT) Tree, 325
  - Application, 94
  - Browser, 316
  - Button Component, 22
  - Button/Check Box/Radio Button, 123
  - Combo Box, 129
  - Component with Text, 28
  - Component with Text Input, 34
  - Figure Canvas, 289
  - Graphics Component, 42
  - HTML Hyperlink, 318
  - Label, 139

- List, 145
- Menu Bar, 164
- Tabbed Pane, 173
- Table, 186
- Text Field/Text Area/Editor Pane/Text Pane, 242
- Toolbar Item, 340
- Tree, 251
- Tree Table, 354
- Component with Text, 28
  - Check Text, 32
  - Store Value, 33
- Component with Text Input, 34
  - Check Editability, 39
  - Input Text, 39
  - Replace Text, 40
- Constants
  - Indices, 373
  - Path Separator, 373
  - Value Separator, 373
- Figure Canvas, 289
  - Check Figure Existence, 297
  - Check Figure Property, 298
  - Check Tool Existence, 300
  - Click Connection, 301
  - Click Figure, 304
  - Click in Figure, 305
  - Drag Figure, 308
  - Drop on Figure, 311
  - Select Tool, 314
- Graphics Component, 42
  - Check Enablement, 45
  - Check Enablement of Context Menu Entry by Indexpath, 46
  - Check Enablement of Context Menu Entry by Indexpath (Specify Position), 47
  - Check Enablement of Context Menu Entry by Textpath, 50
  - Check Enablement of Context Menu Entry by Textpath (Specify Position), 51
  - Check Existence, 54
  - Check Existence of Context Menu Entry by Indexpath, 55
  - Check Existence of Context Menu Entry by Indexpath (Specify Position), 56
  - Check Existence of Context Menu Entry by Textpath, 59

- Check Existence of Context Menu Entry by Textpath (Specify Position), 60
- Check Focus, 63
- Check Property, 64
- Check Selection of Context Menu Entry by Indexpath, 65
- Check Selection of Context Menu Entry by Indexpath (Specify Position), 67
- Check Selection of Context Menu Entry by Textpath, 69
- Check Selection of Context Menu Entry by Textpath (Specify Position), 71
- Click, 74
- Click in Component, 75
- Drag, 77
- Drop, 80
- Select Context Menu Entry by Indexpath, 82
- Select Context Menu Entry by Indexpath (Specify Position), 83
- Select Context Menu Entry by Textpath, 86
- Select Context Menu Entry by Textpath (Specify Position), 88
- Show Text, 91
- Store Property, 92
- Wait for Component, 93
- HTML Hyperlink, 318
  - Check URL, 323
- Indices, 373
- Label, 139
- List, 145
  - Check Existence of Entry by Value, 151
  - Check Selection of Entry by Index/Indices, 152
  - Check Selection of Entry by Value(s), 153
  - Drag Entry by Index, 154
  - Drag Entry by Value, 155
  - Drop on Entry by Index, 158
  - Drop on Entry by Value, 158
  - Select Entry by Index/Indices, 160
  - Select Entry by Value(s), 161
- Menu Bar, 164
  - Check Enablement of Entry by Indexpath, 165
  - Check Enablement of Entry by Textpath, 166
  - Check Existence of Entry by Indexpath, 167
  - Check Existence of Entry by Textpath, 168
  - Check Selection of Entry by Indexpath, 169

- Check Selection of Entry by Textpath, 169
- Select Menu Entry by Indexpath, 171
- Select Menu Entry by Textpath, 171
- Wait for Component, 172

## Path

- relative, 377

Path Separator, 373

Regular Expressions, 15

Relative paths, 377

Simple Match, 19

Tabbed Pane, 173

- Check Enablement of Tab by Index, 179
- Check Enablement of Tab by Value, 180
- Check Existence of Tab, 181
- Check Selection of Tab by Index, 182
- Check Selection of Tab by Value, 182
- Check Text of Tab by Index, 183
- Select Tab by Index, 184
- Select Tab by Value, 185

Table, 186

- Check Editability (Specify Cell), 195
- Check Editability of Cell (Mouse Position), 197
- Check Editability of Selected Cell, 197
- Check Existence of Value in Column, 198
- Check Existence of Value in Row, 200
- Check Text (Mouse Position), 201
- Check Text (Specify Cell), 202
- Drag Cell, 204
- Drag Cell from Column, 209
- Drag Cell from Row, 212
- Drop on Cell, 215
- Drop on Cell from Column, 218
- Drop on Cell from Row, 220
- Input Text (Specify Cell), 222
- Move, 224
- Replace Text (Specify Cell), 227
- Select Cell, 229
- Select Value from Column, 233
- Select Value from Row, 236
- Store Value (Specify Cell), 238
- Store Value at Mouse Position, 240

Text Field/Text Area/Editor Pane/Text Pane, 242

- Insert Text after Index, 247

- Insert Text before/after Pattern, 248
- Select All, 249
- Select Pattern, 249
- Toolbar Item, 340
  - Check Enablement of Entry by Indexpath, 346
  - Check Enablement of Entry by Textpath, 347
  - Check Existence of Entry by Indexpath, 348
  - Check Existence of Entry by Textpath, 349
  - Check Selection of Entry by Indexpath, 350
  - Check Selection of Entry by Textpath, 351
  - Select Menu Entry by Indexpath, 352
  - Select Menu Entry by Textpath, 353
- Tree, 251
  - Check Existence of Node by Indexpath, 258
  - Check Existence of Node by Textpath, 259
  - Check Text (Mouse Position), 261
  - Check Text of Selected Node(s), 262
  - Collapse Node by Indexpath, 263
  - Collapse Node by Textpath, 265
  - Drag Node by Indexpath, 267
  - Drag Node by Textpath, 270
  - Drop on Node by Indexpath, 273
  - Drop on Node by Textpath, 274
  - Expand Node by Indexpath, 276
  - Expand Node by Textpath, 278
  - Move, 280
  - Select Node by Indexpath, 282
  - Select Node by Textpath, 284
  - Store Text at Mouse Position, 287
  - Store Text of Selected Node, 288
- Tree Table, 354
  - Check Selection by Value(s) (Specify Column), 363
  - Select Entry by Indexpath (Specify Column), 364
  - Select Entry by Textpath (Specify Column), 366
- Value Separator, 373