

Clarity

Automated Codeless Test Framework

Version: 8.2.0

User Guide

Table of Contents

DESCRIPTION	3
TECHNICAL REQUIREMENTS.....	3
CLARITY	4
USAGE:.....	4
CONFIGURATION FILE	6
BROWSERSTACK	7
BUSINESSKEYWORDS.....	8
CAPABILITIES.....	9
CONFIG	11
CONTEXT	12
Context.BrowserHeaders.....	15
Context.Capabilities.....	15
Context.SessionSetups.....	15
Context.Requirements.....	15
Context.Defects.....	15
CUSTOMKEYWORDS.....	16
DATABASES	17
DEFECTS	18
DRIVER.....	19
EMAIL	20
GITREPOSITORY.....	21
MANAGEMENT (OBSOLETE)	22
MANUALRESULTS	23
NETWORKS.....	24
OBJECTMAPPINGS	25
REQUIREMENTS.....	26
SLACK.....	27
SESSIONSETUPS.....	28
TESTEXCEPTIONS	29
TESTSCENARIOS	30
DATA	31
ITEM DEFINITION	32
BUSINESS KEYWORD.....	32

CUSTOM KEYWORD.....	33
DATABASE.....	34
DEFECT.....	35
NETWORK.....	36
OBJECT MAPPING.....	37
<i>MultipleContent</i>	38
<i>Advanced Object Mapping</i>	38
REQUIREMENT.....	40
SESSION SETUP.....	41
TEST SCENARIO.....	42
<i>Test Results</i>	42
<i>[Scenario] section</i>	43
<i>Test Steps</i>	43
<i>Test Data</i>	44
TEST EXCEPTION.....	46
DATA.....	47
<i>User-defined Data</i>	47
<i>Local and Global Data</i>	47
<i>Data Usage</i>	47
<i>MultipleValue</i>	48
<i>Technical Data</i>	48
<i>Technical Data for Capabilities</i>	49
<i>Data Type: TextFile</i>	49
<i>Data Type: BinaryFile</i>	49
<i>Data Type: URL</i>	49
<i>Data Type: Table</i>	49
DATA PATTERN.....	50
<i>Default Format</i>	51
<i>Data Pattern Examples</i>	51
APPENDIX A – FUNCTIONAL KEYWORDS.....	52
ACTIONS.....	52
VALIDATIONS.....	55
APPENDIX B – EMAIL KEYWORDS.....	58
ACTIONS.....	58
VALIDATIONS.....	58
APPENDIX C – API KEYWORDS.....	59
ACTIONS.....	59
VALIDATIONS.....	60
APPENDIX D – VISUAL KEYWORDS.....	61
ACTIONS.....	61
VALIDATIONS.....	62
APPENDIX E – FLOW KEYWORDS.....	63
LOGIC.....	63
LOOP.....	63
EXAMPLES.....	63

APPENDIX F – KEYWORDS DATA	65
APPENDIX G – OPTIONS	66
SYSTEM OPTIONS	66
CUSTOM OPTIONS	68
LOOP OPTIONS.....	68
OPTION: TRANSFORM	69
APPENDIX H – KEYS CHARACTERS	70
APPENDIX I – SAFARI CONFIGURATION	72
APPENDIX J – LOCAL MOBILE CONFIGURATION	73
APPENDIX K – PATH MANAGEMENT	74
APPENDIX L – CUSTOM KEYWORD DEVELOPMENT	75
METHOD SIGNATURE	75
METHOD RETURN TYPE & VALUE	75
EXAMPLE	76
APPENDIX M – DATA TRANSFORMATION	77
EXAMPLES: STRING TRANSFORMATION	77
EXAMPLES: NUMBER TRANSFORMATION.....	77
EXAMPLES: DATE TRANSFORMATION.....	78
EXAMPLES: TRANSFORMATION FUNCTIONS	78

Description

Clarity is an automated codeless test framework allowing testers to execute automated tests driven by test scenarios without the need for code.

Technical Requirements

- *Java JDK 1.8+*
- *Browser/Mobile Testing: Selenium drivers*

Clarity

Usage:

Java -jar Clarity [list of options]

Options	Parameter	Option Dependency	Description
-Config	Configuration File		Configuration. Defines the configuration file to use for the test execution.
-NoLogConsole		-Config	No Log Console. Sets the log information not to be displayed on the console.
-Url		-Config	URL. Defines the default URL to use for the test execution.
-Name		-Config	Name. Defines a name to be associated to the test execution.
-Version			Version. Returns the version of the Clarity framework.
-ID			ID. Returns Clarity Identifier
-UserStamp			User Stamp. Returns a Clarity user stamp to identify user input.
-ReqCoverage		-Config	Requirement Coverage. Generates statistics on the requirement risk analysis and test coverage. This option requires the definition of requirements and the mapping between test scenarios and requirements.
-GitConnect		-Config	Git Connection. Connects to a remote Git Repository.
-GitSync		-Config	Git Synchronization. Connects to a Git Repository and synchronizes the local and remote repositories without running any test.
-ReportGen	JSON Report		Report Generation. Generates a HTML version of the JSON execution report using the default report template.
-ReportTemplate	Report Template File		Custom report template file to use for report generation.
-SafeExec	Low High Headless	-Config	Safe Execution. Safe mode execution. The framework runs each test until getting 2 identical consecutive results, and up to a maximum of 5 runs. If the "Low" parameter is used only one run is considered is the result is "Passed". If the "Headless" parameter is used, the framework runs in Headless mode and only switch to the targeted capability if a failure occurs.
-Consolidate	Consolidation Folder	-Config	Consolidate Run. Consolidates the results of multiple runs into a single consolidated run result. The consolidation folder should contains the runs JSON files in the main folder or in subfolders.

-Delta	Consolidation Folder	-Config	Delta Run. Performs a delta run based on the results of a consolidation process (as described for the -Cons option) Only the tests that have not passed will be re-executed.
-DeltaFromLast		-Config	Delta From Last. Performs a delta run from the last run results.
-DeltaLevel	High Low (Default)	-Delta -DeltaFromLast	Delta Level. Performs a delta run based on the results of a consolidation process (as described for the -cons option) Only the tests that have not passed will be re-executed. The selection of the tests will be done using the following rules: <ul style="list-style-type: none"> • DeltaLevel = Low, Failed and Not-Executed tests will be processed. • DeltaLevel = High, all tests with a status different from Passed will be processed
-QA	Test (Default) Other	-Config	Quality Assurance. Validate the quality of the test scenario. Based on the option parameter, the validations performed are: <ul style="list-style-type: none"> • Test: Validate that the last step of a test is a validation step • Other: No validation
-SkipIfSameScope			Skip If Same Scope. Clarity will not run the test configuration if the list of tests, requirements and defects are identical from the last execution.

Configuration File

The configuration file is a YAML file defining the different components to use during the test execution. The configuration file contains the following sections:

Section	Description
BrowserStack	<i>Defines the configuration attribute to use for BrowserStack integration</i>
BusinessKeywords	<i>Define the list of business keyword files to use for the test execution</i>
CustomKeywords	<i>Define the list of custom keyword files to use for the test execution</i>
Capabilities (*)	<i>Defines the client platforms (Capability) to use for the test execution</i>
Config	<i>Defines the technical configuration parameters for the test execution</i>
Context (*)	<i>Defines the driver configuration for the test execution</i>
Databases	<i>Defines the list of databases to use for the test execution</i>
Defects	<i>Defines the list of defect files to use for the test scenario mapping</i>
Email	<i>Defines the Email configuration to use for sending email</i>
GitRepository	<i>Defines the Git configuration for the test execution</i>
Management	<i>Defines the information needed to manage regular sets of tests, requirements and defects (ex. Regression packs)</i>
ManualResults	<i>Defines the list of manual result files to use for the test execution</i>
Networks	<i>Defines the list of network files to use for the test execution</i>
ObjectMappings	<i>Defines the list of object mapping files to use for the test execution</i>
Requirements	<i>Defines the list of requirement files to use for the test scenario mapping</i>
SessionSetups	<i>Defines the list of session setup files to use for the test execution</i>
Slack	<i>Defines the Slack App configuration to use for slack integration</i>
TestExceptions	<i>Define the list of test exception files to use for the test execution</i>
TestScenarios (*)	<i>Defines the list of test scenario files to use for the test execution</i>
Data	<i>Defines the list of data files to use for the test execution</i>

* Mandatory section

BrowserStack

The BrowserStack section contains a list of attributes needed to integrate with the application BrowserStack.

Attribute	Values	Description
Username*		<i>Defines the username to use for connecting to BrowserStack.</i>
AccessKey*		<i>Defines the file path than contains the accesskey to use for connecting to BrowserStack.</i>
Video	<i>True (Default) False</i>	<i>Defines if a video should be generated by BrowserStack.</i>
NetworkTunnel	<i>True False (Default)</i>	<i>Defines if a network tunnel should be created when connecting to BrowserStack</i>

* Mandatory attribute

BusinessKeywords

The *BusinessKeywords* section contains a list of business keyword configurations. Each configuration contains the following attributes:

Attribute	Values	Description
Path*		<i>Defines the path where the business keyword files are stored.</i>
IncludeSubFolder	<i>True False (Default)</i>	<i>Defines if the test framework should consider sub folder from the attribute "Path" when retrieving the business keyword files.</i>
Filter		<i>Defines a regular expression filter to apply to the list of business keyword files identified from the path folder. All files satisfying the regular expression will be considered by the test framework. This attribute can contain a comma separated list of values.</i>
ExcludeFilter		<i>Defines a regular expression filter to apply to the list of business keyword files identified from the path folder. All files satisfying the regular expression will be discarded by the test framework. This attribute can contain a comma separated list of values.</i>

* *Mandatory attribute*

Capabilities

The Capabilities section contains a list of capability configurations. Each configuration contains the following attributes:

Attribute	Values	Description
Name*		Defines the name of the capability. The name should only contain numbers, letters, "-", "_" and ".".
Type*	API Browser LocalDevice Emulator BrowserStack	Defines the type of capability
MaxSession	Integer > 0 1 (Default)	Defines the maximum number of sessions to use for the capability.
Browser*		Defines the data structure defining the browser associated with the capability. This attribute is mandatory when: <ul style="list-style-type: none"> Attribute Type = Browser
LocalDevice*		Defines the data structure defining the real mobile associated with the capability. This attribute is mandatory when: <ul style="list-style-type: none"> Attribute Type = LocalDevice
Emulator*		Defines the data structure defining the emulator associated with the capability. The browser and mobile emulation is performed using the Chrome browser features. This attribute is mandatory when: <ul style="list-style-type: none"> Attribute Type = Emulator
BrowserStack*		Defines the browserstack configuration associated with the capability. This attribute is mandatory when: <ul style="list-style-type: none"> Attribute Type = BrowserStack

* Mandatory attribute

Browser Attribute	Values	Description
Name*	Headless Chrome Firefox Safari Edge	Defines the type of browser associated with the capability. The headless type requires the Chrome driver to be defined.
Version		Defines the version of the browser associated with the capability
Download		Defines the folder of the capability where all downloads will be stored. Property available for: Headless, Chrome and Firefox browsers.
Width	Integer > 0 Max (Default)	Defines the width of the capability. If the value Max is used, the framework will set the capability to its maximum width and height.
Height	Integer > 0 Max (Default)	Defines the height of the capability. If the value Max is used, the framework will set the capability to its maximum width and height.
DriverPath*		Defines the path for the driver associated with the selected browser type.
BinaryPath*		Defines the path for the binary file associated with the selected browser type.
HeadlessDriverPath		Defines the path of the chrome driver to use in headless mode when safe runs are configured.
HeadlessBinaryPath		Defines the path for the chrome binary file to use in headless mode when safe runs are configured.

* Mandatory attribute

LocalDevice Attribute	Values	Description
DeviceType*	Mobile Tablet	Defines the type of the device associated with the capability.
DeviceID*		Defines the ID of the device associated with the capability.
DeviceName*		Defines the name of the device associated with the capability.
DeviceURL*		Defines the URL of the device associated with the capability.
Orientation	Portrait (Default) Landscape	Defines the orientation associated with the capability
BrowserName*		Defines the name of the browser to use on the device.
OS*	Android iOS	Defines the Operating System of the device associated with the capability.
OSVersion		Defines the version of the operating system of the device associated with the capability.

* Mandatory attribute

Emulator Attribute	Values	Description
Type*	Headless Chrome	Defines the type of emulator associated with the capability
Download		Defines the folder of the capability where all downloads will be stored.
Width*	Integer > 0	Defines the width of the capability.
Height*	Integer > 0	Defines the height of the capability.
UserAgent*		Defines the user agent information associated with the capability.
Orientation	Portrait (Default) Landscape	Defines the orientation associated with the capability
PixelRatio*	Double > 0 1.0 (Default)	Defines the pixel ratio associated with the capability
ChromeDriverPath*		Defines the path for the Chrome driver.
ChromeBinaryPath*		Defines the path for the Chrome binary file.
HeadlessDriverPath		Defines the path of the chrome driver to use in headless mode when safe runs are configured.
HeadlessBinaryPath		Defines the path for the chrome binary file to use in headless mode when safe runs are configured.

* Mandatory attribute

BrowserStack Attribute	Values	Description
DeviceType*	Browser Mobile Tablet	Defines the device type associated with the capability
OS*	DeviceType = Browser <ul style="list-style-type: none"> OSX Windows DeviceType = Mobile <ul style="list-style-type: none"> Android iOS DeviceType = Tablet <ul style="list-style-type: none"> Android iOS 	Defines the Operating System associated with the capability
OSVersion*		Defines the Operating System version associated with the capability
Resolution*		Defines the resolution associated with the capability. This attribute is mandatory when: <ul style="list-style-type: none"> DeviceType = Browser
BrowserName*	Chrome Firefox Safari IE Edge Opera	Defines the browser name associated with the capability. This attribute is mandatory when: <ul style="list-style-type: none"> DeviceType = Browser
BrowserVersion*		Defines the browser version associated with the capability. This attribute is mandatory when: <ul style="list-style-type: none"> DeviceType = Browser
DeviceName*		Defines the device name associated with the capability. This attribute is mandatory when: <ul style="list-style-type: none"> DeviceType = Mobile or Tablet
RealEquipment	True (Default) False	Defines if a real equipment is used to support the capability.
Orientation	Portrait (Default) Landscape	Defines the orientation associated with the capability

* Mandatory attribute

Config

The Config section contains the following attributes

Attribute	Values	Description
LogLevel	All Debug Error Fatal Info Off Trace Warn (Default)	Defines the level of the log to use for the test execution.
LogToConsole	True (Default) False	Defines if the log information should also be displayed on the console.
FixQuantityTest	Integer >= 0	Defines the number of tests to execute. The tests are selected randomly from the target list defined in the TestScenarios section. If the value is equal to 0, no filtering is performed.
FixQuantityCase	Integer >= 0	Defines the number of test instances to execute. The test instances are selected randomly from the target list defined in the TestScenarios section. If the value is equal to 0, no filtering is performed.
FixCaseNumber	Integer >= 0	Defines the instance number of the test to execute. The test instance is selected from the target list defined in the TestScenarios section. If the value is equal to 0, no filtering is performed. If FixCaseNumber is greater than the number of instances of a test, the first instance is selected.
ReferencePath		Defines the reference file path to be used by the framework
License	License.key (Default)	Defines the name of the file containing Clarity license
LicenseServer		Defines the URL of the Clarity license server.

Context

The Context section contains the following attributes

Attribute	Values	Description
ConfigName		Defines the name for the configuration
DeliveryCode		Defines a code identifying the system under test
Timeout	Integer > 0 30000 (Default)	Defines the timeout (in milliseconds) to use by the driver during the test execution.
ResultFolder	Any Results (Default)	Defines the result folder where the test execution results will be stored.
ExportResult	True False (Default)	Defines if the execution details are exported after the test execution.
ExportFolder		Defines the folder where results will be copied after the test execution. This attribute is mandatory when ExportResult = True
ScreenCapture	None All NotPassed Failed (Default)	Defines if the test execution will include the capture of screenshots. <ul style="list-style-type: none"> • None: No screenshot is taken • All: Screenshots are taken before and after each test step • Failed: Screenshots are taken only if a step status is equal to: Failed • NotPassed: Screenshots are taken only if a step status is not equal to: Passed
ScreenCaptureOnTestException	True False (Default)	Defines if the test execution will include the capture of screenshots when test exceptions are executed.
CloseAfterRun	True (Default) False	Defines if the client platform (browser, app, etc..) should be close after the test execution.
DefaultURL		Defines the value of the default URL to use for the test execution.
SkipIfSameScope		Defines if Clarity will not run the test configuration if the list of tests, requirements and defects are identical from the last execution.
TestSharding	True False (Default)	Defines if test execution will automatically distribute the test scenarios across the different capabilities. If the value is false, all test scenarios will be executed for each defined capability.
RunName		Defines the name of the run to use for the test execution report. The default run name is: Clarity
MaxSession	Integer > 0 1 (Default)	Defines the maximum number of sessions (clients) to use simultaneously during the test execution.
Tag		Defines a logical expression used to filter the test scenario list to consider for the test execution.
VisualTolerance	Integer >= 0	Defines the tolerance acceptable for visual validations. The value is expressed in pixels.
SafeRunOnDisk	True False (Default)	Defines if the safe runs are saved on disk
PartialResultOnDisk	Integer > 0	Defines if partial results are saved on disk. The value indicates the interval (in minutes) between each partial results.
AddHoursToUTC	Integer 0 (Default)	Defines the number of hours to be added to the default Clarity UTC time zone.
ExportSafeRun	True False (Default)	Defines if the safe run execution details are exported after the test execution.
ExportPartialResult	True False (Default)	Defines if the partial result execution details are exported after the test execution.

BrowserHeaders		List of headers to be added by to the selected capability of type "Browser".
EmailResultTo		Defines the list of emails (comma separated) to use for sending the test execution results.
EmailResultSubject		Defines the text of the subject to use when sending the test execution results by email.
EmailReportTo		Defines the list of emails (comma separated) to use for sending the test execution HTML report.
EmailReportSubject		Defines the text of the subject to use when sending the test execution HTML report by email.
EmailReportMaxSize	Number > 0 10 (Default)	Defines the maximum size (in Mb) for the HTML Report authorised to be sent by email.
EmailFrom		Email to use as the sender of the emails.
DisplayEmailFrom		Name of the sender to be displayed to the email receivers.
EmailReplyTo		List of emails (comma separated) to consider when replying to the sent emails.
SlackResultUserTo		Defines the list of emails (comma separated) from the Slack users to use for sending the test execution results.
SlackResultChannelTo		Defines the list of Slack channels to use for sending the test execution results.
SlackResultSubject		Defines the text of the subject to use when sending the test execution results using Slack.
SlackReportUserTo		Defines the list of emails (comma separated) from the Slack users to use for sending the test execution HTML report.
SlackReportChannelTo		Defines the list of Slack channels to use for sending the test execution HTML report.
SlackReportSubject		Defines the text of the subject to use when sending the test execution HTML report using Slack.
SlackReportMaxSize	Number > 0 10 (Default)	Defines the maximum size (in Mb) for the HTML Report authorised to be sent using Slack.
ReportTemplate		Defines the path of the report template
CompactReportTemplate		Defines the path of the compact report template. The compact report template is used automatically when the number of test execution instances is greater than 250
ImageComparisonLevel	Exact Strict (Default) Tolerant VeryTolerant	Defines how 2 images are compared: <ul style="list-style-type: none"> Exact: pixel to pixel comparison Strict: compares everything including content (text), fonts, layout, colors and position of each of the elements. Strict knows to ignore rendering changes that are not visible to the human (anti-aliasing changes, small pixel movements and various other changes that are typically caused when running tests on different machines with different graphic cards, etc.) Tolerant: accepts changes which are still visible, but barely VeryTolerant: accepts larger changes than the Tolerant level
SaveImageCapture	Never (Default) FailedCheck Always	Defines when the image capture (object or page) are saved on disk: <ul style="list-style-type: none"> Never: Image capture is never saved FailedChecked: Image captured is saved when an image comparison fails.

		<ul style="list-style-type: none"> Always: Image capture is saved anytime an image comparison is performed.
SaveImageCaptureFolder		<p>Defines the folder where image captures will be saved..</p> <p>This attribute is mandatory when SaveImageCapture is different from Never</p>
ContinueOnFailure	<p>True</p> <p>False (Default)</p>	<p>Defines if test execution should continue if the result of a step is not equal to PASSED</p>
AllowedConsecutiveFailedSteps	<p>Integer > 0</p>	<p>Defines how many consecutive failed steps are allowed when ContinueOnFailure = True</p>
NetworkSimulation		<p>Defines the network configuration to use for the test execution</p>
ForceProxyActivation	<p>True</p> <p>False (Default)</p>	<p>Forces a proxy to be used for the test execution</p>

Context.BrowserHeaders

The Context.BrowserHeaders attribute contains a list of header definition using the following attributes:

Attribute	Values	Description
Name*		Name of the header
Value		Value of the header

* Mandatory attribute

Context.Capabilities

The Context.Capabilities attribute contains the list of capabilities to use for the test execution:

Attribute	Values	Description
Name*		Name of the capability
MaxSession		Maximum number of session for the capability
Orientation	Portrait (Default) Landscape	Define the orientation to use for mobile capabilities.

* Mandatory attribute

Context.SessionSetups

The Context.SessionSetups attribute contains the list of session setup references to use for the test execution. Session setups are defined as a list of steps to be executed before each test scenario.

Attribute	Values	Description
Name*		Name of the session setup configuration

* Mandatory attribute

Context.Requirements

The Context.Requirements attribute contains the list of requirement systems to use for the test execution:

Attribute	Values	Description
System*		Name of the requirement system

* Mandatory attribute

Context.Defects

The Context.Defects attribute contains the list of defect systems to use for the test execution:

Attribute	Values	Description
System*		Name of the defect system

* Mandatory attribute

CustomKeywords

The CustomKeywords section contains a list of custom keyword configurations. Each configuration contains the following attributes:

Attribute	Values	Description
Path*		<i>Defines the path where the custom keyword files are stored.</i>
IncludeSubFolder	<i>True False (Default)</i>	<i>Defines if the test framework should consider sub folder from the attribute "Path" when retrieving the custom keyword files.</i>
Filter		<i>Defines a regular expression filter to apply to the list of custom keyword files identified from the path folder. All files satisfying the regular expression will be considered by the test framework. This attribute can contain a comma separated list of values.</i>
ExcludeFilter		<i>Defines a regular expression filter to apply to the list of custom keyword files identified from the path folder. All files satisfying the regular expression will be discarded by the test framework. This attribute can contain a comma separated list of values.</i>

* Mandatory attribute

Databases

The Databases section contains a list of database configurations. Each configuration contains the following attributes:

Attribute	Values	Description
Path*		<i>Defines the path where the database files are stored.</i>
IncludeSubFolder	<i>True False (Default)</i>	<i>Defines if the test framework should consider sub folder from the attribute "Path" when retrieving the database files.</i>
Filter		<i>Defines a regular expression filter to apply to the list of database files identified from the path folder. All files satisfying the regular expression will be considered by the test framework. This attribute can contain a comma separated list of values.</i>
ExcludeFilter		<i>Defines a regular expression filter to apply to the list of database files identified from the path folder. All files satisfying the regular expression will be discarded by the test framework. This attribute can contain a comma separated list of values.</i>

* Mandatory attribute

Defects

The Defects section contains a list of defect configurations. Each configuration contains the following attributes:

Attribute	Values	Description
Path*		<i>Defines the path where the defects files are stored.</i>
IncludeSubFolder	<i>True False (Default)</i>	<i>Defines if the test framework should consider sub folder from the attribute "Path" when retrieving the test scenario files.</i>
Filter		<i>Defines a regular expression filter to apply to the list of defects files identified from the path folder. All files satisfying the regular expression will be considered by the test framework. This attribute can contain a comma separated list of values.</i>
ExcludeFilter		<i>Defines a regular expression filter to apply to the list of defects files identified from the path folder. All files satisfying the regular expression will be discarded by the test framework. This attribute can contain a comma separated list of values.</i>

* Mandatory attribute

Driver

The Driver section contains the following attributes

Attribute	Values	Description
Type*	Selenium	Defines the type of execution driver to use for the test execution. Possible value: Selenium

* Mandatory attribute

Email

The Email section contains a list of attributes needed to integrate with a SMTP server in order to send emails.

Attribute	Values	Description
SMTPHost		<i>Defines the host name associated with the SMTP Server.</i>
SMTPPort	<i>Number > 0 25 (Default)</i>	<i>Defines the port to use for connecting to the SMTP Server.</i>
SMTPUsername		<i>Defines the username to use for connecting to the SMTP Server.</i>
SMTPPassword		<i>Defines the file path than contains the password to use for connecting to the SMTP Server.</i>
SMPTPEncryption	<i>None (Default) TLS SSL</i>	<i>Defines the security mode to use for connecting to the SMTP Server.</i>
POP3Host		<i>Defines the host name associated with the POP3 Server.</i>
POP3Port	<i>Number > 0 110 (Default)</i>	<i>Defines the port to use for connecting to the POP3 Server.</i>
POP3Username		<i>Defines the username to use for connecting to the POP3 Server.</i>
POP3Password		<i>Defines the file path than contains the password to use for connecting to the POP3 Server.</i>
POP3Encryption	<i>None (Default) TLS SSL</i>	<i>Defines the security mode to use for connecting to the SMTP Server.</i>
InboxFolder		<i>Defines the name of the folder from which emails are retrieved.</i>

* Mandatory attribute

GitRepository

The *GitRepository* section contains the following attributes

Attribute	Values	Description
RemoteURL*		<i>Defines the git URL to the remote repository.</i>
LocalFolder*		<i>Defines the path to the local repository. If the path does not exist, the framework will clone the remote repository.</i>
Username*		<i>Defines the username to use to connect to the remote repository.</i>
Password*		<i>Defines the path to a file that contains the password to connect to the remote repository.</i>
ReferencePath		<i>Defines the path to the Git local directory to consider as the reference for the test execution.</i>
ResultFolder		<i>Defines the path to the Git local directory where the test execution results will be saved.</i>
PushOnExit	<i>True False (Default)</i>	<i>Defines if the changes of the repository content should be automatically pushed to the remote repository.</i>
SyncOnStart	<i>True (Default) False</i>	<i>Defines if the repository content should be synchronized automatically with the remote repository before starting the test execution.</i>

* Mandatory attribute

Management (Obsolete)

Note: This section has been kept for compatibility reasons but will be removed in the next releases.

The Management section contains a list of attributes needed to manage regular sets of tests, requirements and defects (ex. Regression packs)

Attribute	Values	Description
Users		Defines the path to a TSV (tab separated) file with the list of users of the framework.
Data		Defines the path to a TSV (tab separated) file with the list of predefined data.
ExecGroups		Defines the path to a TSV (tab separated) file with the list of predefined execution groups and the associated data values.
DataFolder*		Defines the directory where management data is stored
System*		Defines a unique identifier for the system under test.

* Mandatory attribute

ManualResults

The ManualResults section contains a list of test scenario configurations used for manual execution. Each configuration contains the following attributes:

Attribute	Values	Description
Path*		<i>Defines the path where the test scenario files are stored.</i>
IncludeSubFolder	<i>True False (Default)</i>	<i>Defines if the test framework should consider sub folder from the attribute "Path" when retrieving the test scenario files.</i>
Filter		<i>Defines a regular expression filter to apply to the list of test scenario files identified from the path folder. All files satisfying the regular expression will be considered by the test framework. This attribute can contain a comma separated list of values.</i>
ExcludeFilter		<i>Defines a regular expression filter to apply to the list of test scenario files identified from the path folder. All files satisfying the regular expression will be discarded by the test framework. This attribute can contain a comma separated list of values.</i>

* Mandatory attribute

Networks

The Networks section contains a list of network configurations. Each configuration contains the following attributes:

Attribute	Values	Description
Path*		<i>Defines the path where the network files are stored.</i>
IncludeSubFolder	<i>True False (Default)</i>	<i>Defines if the test framework should consider sub folder from the attribute "Path" when retrieving the network files.</i>
Filter		<i>Defines a regular expression filter to apply to the list of network files identified from the path folder. All files satisfying the regular expression will be considered by the test framework. This attribute can contain a comma separated list of values.</i>
ExcludeFilter		<i>Defines a regular expression filter to apply to the list of network files identified from the path folder. All files satisfying the regular expression will be discarded by the test framework. This attribute can contain a comma separated list of values.</i>

* Mandatory attribute

ObjectMappings

The *ObjectMappings* section contains a list of object mapping configurations. Each configuration contains the following attributes:

Attribute	Values	Description
Path*		<i>Defines the path where the object mapping files are stored.</i>
IncludeSubFolder	<i>True False (Default)</i>	<i>Defines if the test framework should consider sub folder from the attribute "Path" when retrieving the object mapping files.</i>
Filter		<i>Defines a regular expression filter to apply to the list of object mapping files identified from the path folder. All files satisfying the regular expression will be considered by the test framework. This attribute can contain a comma separated list of values.</i>
ExcludeFilter		<i>Defines a regular expression filter to apply to the list of object mapping files identified from the path folder. All files satisfying the regular expression will be discarded by the test framework. This attribute can contain a comma separated list of values.</i>

* *Mandatory attribute*

Requirements

The Requirements section contains a list of requirement configurations. Each configuration contains the following attributes:

Attribute	Values	Description
Path*		<i>Defines the path where the requirements files are stored.</i>
IncludeSubFolder	<i>True False (Default)</i>	<i>Defines if the test framework should consider sub folder from the attribute "Path" when retrieving the test scenario files.</i>
Filter		<i>Defines a regular expression filter to apply to the list of requirements files identified from the path folder. All files satisfying the regular expression will be considered by the test framework. This attribute can contain a comma separated list of values.</i>
ExcludeFilter		<i>Defines a regular expression filter to apply to the list of requirements files identified from the path folder. All files satisfying the regular expression will be discarded by the test framework. This attribute can contain a comma separated list of values.</i>

* Mandatory attribute

Slack

The Slack section contains a list of attributes needed to integrate with the application Slack.

Attribute	Values	Description
Token*		Defines the file path than contains the token to use for connecting to Slack.

* Mandatory attribute

The token can be associated with a Slack user or with a Slack app (bot). In both configurations, the access right privileges should include:

- *users:read*
- *users:read:emai*
- *chat:write*
- *files:write*

SessionSetups

The SessionSetups section contains a list of session setup configurations. Each configuration contains the following attributes:

Attribute	Values	Description
Path*		<i>Defines the path where the session setup files are stored.</i>
IncludeSubFolder	<i>True False (Default)</i>	<i>Defines if the test framework should consider sub folder from the attribute "Path" when retrieving the session setup files.</i>
Filter		<i>Defines a regular expression filter to apply to the list of session setup files identified from the path folder. All files satisfying the regular expression will be considered by the test framework. This attribute can contain a comma separated list of values.</i>
ExcludeFilter		<i>Defines a regular expression filter to apply to the list of session setup files identified from the path folder. All files satisfying the regular expression will be discarded by the test framework. This attribute can contain a comma separated list of values.</i>

* Mandatory attribute

TestExceptions

The TestExceptions section contains a list of test exception configurations. Each configuration contains the following attributes:

Attribute	Values	Description
Path*		<i>Defines the path where the test exception files are stored.</i>
IncludeSubFolder	<i>True False (Default)</i>	<i>Defines if the test framework should consider sub folder from the attribute "Path" when retrieving the test exception files.</i>
Filter		<i>Defines a regular expression filter to apply to the list of test exception files identified from the path folder. All files satisfying the regular expression will be considered by the test framework. This attribute can contain a comma separated list of values.</i>
ExcludeFilter		<i>Defines a regular expression filter to apply to the list of test exception files identified from the path folder. All files satisfying the regular expression will be discarded by the test framework. This attribute can contain a comma separated list of values.</i>

* Mandatory attribute

TestScenarios

The TestScenarios section contains a list of test scenario configurations. Each configuration contains the following attributes:

Attribute	Values	Description
Path*		<i>Defines the path where the test scenario files are stored.</i>
IncludeSubFolder	<i>True False (Default)</i>	<i>Defines if the test framework should consider sub folder from the attribute "Path" when retrieving the test scenario files.</i>
Filter		<i>Defines a regular expression filter to apply to the list of test scenario files identified from the path folder. All files satisfying the regular expression will be considered by the test framework. This attribute can contain a comma separated list of values.</i>
ExcludeFilter		<i>Defines a regular expression filter to apply to the list of test scenario files identified from the path folder. All files satisfying the regular expression will be discarded by the test framework. This attribute can contain a comma separated list of values.</i>

* Mandatory attribute

Data

The Data section contains a list of data configurations. Each configuration contains the following attributes:

Attribute	Values	Description
Path*		<i>Defines the path where the data files are stored.</i>
IncludeSubFolder	<i>True False (Default)</i>	<i>Defines if the test framework should consider sub folder from the attribute "Path" when retrieving the data files.</i>
Filterk		<i>Defines a regular expression filter to apply to the list of data files identified from the path folder. All files satisfying the regular expression will be considered by the test framework. This attribute can contain a comma separated list of values.</i>
ExcludeFilter		<i>Defines a regular expression filter to apply to the list of data files identified from the path folder. All files satisfying the regular expression will be discarded by the test framework. This attribute can contain a comma separated list of values.</i>

* Mandatory attribute

Item Definition

Business Keyword

The Business Keyword file is a YAML file containing a list of business keyword definitions. A business keyword is a way to group standard keywords in order to perform a business operation.

The attributes to define the business keyword are the following:

Attribute	Description
Name*	Defines a name for the business keyword. The name should only contain numbers, letters, "-", "_" and ".".
RegEx*	Defines a regular expression that the framework will use to associate a test step with a created business keyword
Steps	Defines of list of steps associated with the business keywords. A step can be either a standard keyword or another business keyword. Within a business keyword, step data can be linked to the business keyword step data by using a reference to the business keyword regex group. For instance [1] will use the value of the first regex group. The property is mandatory if the Path property is not defined
Path	Defines the path of a text file containing the business keyword steps. The property is mandatory if the Steps property is not defined

* Mandatory attribute

Custom Keyword

The Custom Keyword file is a YAML file containing a list of custom keyword definitions. A custom keyword is a way to create new bespoke keywords usable by the framework.

The attributes to define the custom keyword are the following:

Attribute	Values	Description
RegEx*		Defines a regular expression that the framework will use to associate a test step with a created custom keyword
Platform	ALL WEB	Defines the platform associated with the custom keyword
ObjectIndex		Comma separated list of integer mapping the group of the regex that contains an object name. If the platform is different than WEB, this attribute is ignored.
LibraryPath		Define the path to the Java JAR file containing the logic of the custom keyword.
ClassName		Defines the full name of the Java Class containing the logic of the custom keyword.
MethodName		Defines the name of the Java Method containing the logic of the custom keyword.

* Mandatory attribute

Database

The Database file is a YAML file containing a list of database definitions. This information is used to connect to a database instance.

The attributes to define the database are the following:

Attribute	Values	Description
Name*		Defines a name for database. The name should only contain numbers, letters, "-", "_" and ".".
Type*	SQLite	Defines the database type
MaxSessions	Number > 0 1 (Default)	Defines the maximum number of concurrent sessions allowed when connecting to the database
Path		Defines the path to the database file associated with the database. This attribute is mandatory when: Attribute Type = SQLite
Username		Defines the username to use to connect to the database
Password		Defines the path to a file that contains the password to connect to the database

* Mandatory attribute

Defect

The defect file is a YAML file containing a list of defect information. This information is used to map the test scenarios and therefore compute the defect test coverage.

The attributes to define the defect are the following:

Attribute	Values	Description
Name*		Defines the name of the defect The name should only contain numbers, letters, "-", "_" and ".".
System*		Defines the system associated with the defect.
Description		Defines the description of the defect.
Origin*		Defines the origin of the defect (ex. prod, release, etc..)
Severity*	Number ≥ 1 and ≤ 5	Defines the severity associated with the defect
FixedIn		Defines the reference to the release that has fixed the defect
Capability		Defines the name of the capability associated with the defect

* Mandatory attribute

Network

The network file is a YAML file containing a list of network items. A network definition can be chosen in order to limit the network bandwidth and therefore simulate real network behaviour.

The attributes to define the network configuration are the following:

Attribute	Values	Description
Name*		Defines a name for the network item. The name should only contain numbers, letters, "-", "_" and ".".
DownloadSpeed*	Number (>0)	Defines the download speed of the network item. The download speed is expressed in Mbps (Megabit per second)
UploadSpeed*	Number (>0)	Defines the upload speed of the network item. The upload speed is expressed in Mbps (Megabit per second)

* Mandatory attribute

Object Mapping

The Object Mapping file is a YAML file containing a list of mappings between the name of the UI objects (logical name) used in the test steps and the technical configuration that allows the automaton engine to localise the object.

The attributes to define the object mapping are the following:

Attribute	Values	Description
Name*		Defines the logical name of the object. The name should only contain numbers, letters, "-", "_" and ".".
Type*	Css ID Class Name XPath	Defines the type of mapping to use for the object. XPath is not allowed when mapping child objects of a Shadow object.
Value*		Defines the value of the technical mapping for the object.
Index	First (Default) Last Integer > 0	Defines the index of the object when type and value attributes identify more than one object.
Content		Defines the value of the content of the object when type and value attributes identify more than one object. If content starts with the character "!", this property defines the value that a object should not contain.
Selected	True False	Defines if the object is selected (ex. Checkbox, radio button)
MultipleContent		Defines the name of the attribute of the object to consider as content.
Depth	Integer > 0	Defines the hierarchy depth, compared to the parent or container object, where the object should be identified.
Container		Defines the logical name of the container object where the object is located.
Parent		Defines the logical name of the parent object of the object.
ParentIndex	First Last Integer > 0	Defines the index of the parent when clarity finds more than one parent. If the attribute ParentIndex is omitted, Clarity will use the value of the parent object index attribute.
Sibling		Defines the logical name of a sibling object that belongs to the same parent or container.
SiblingDepth	Integer > 0	Defines the hierarchy depth, compared to the parent object, where the sibling should be identified.
Child		Defines the logical name of a child object.
ChildDepth	Integer > 0	Defines the hierarchy depth, compared to the object, where the child should be identified.
AboveOf		Defines the logical name of an object that is localized below the targeted object
BelowOf		Defines the logical name of an object that is localized above the targeted object
RightOf		Defines the logical name of an object that is localized left of the targeted object
LeftOf		Defines the logical name of an object that is localized right of the targeted object
NearOf		Defines the logical name of an object that is localized near the targeted object
Hidden	True False	Defines if the object is a hidden object (ex. Object part of the Head section of a page, input object of type file, hidden visibility, etc...)

* Mandatory attribute

MultipleContent

In order to associate multiple content to an object, the attribute `MultipleContent` can be used. In this case, the attribute contains the name of the attribute to be used as the object content.

For example, in a Multilanguage system, the content of an object can contain a value translated in each language. In order to select the right content based on the country the following object definition can be used:

```
- Name: Account.CreateAccount
  Type: css
  Value: "a"
  MultipleContent: Content_{CountryCode}
  Content_US: "Create an Account"
  Content_JP: "アカウントを登録する"
```

If the data `CountryCode` contains the value "US", the attribute `Content_US` will be used as the object content. If the `CountryCode` is equal to "JP", the framework will use instead the attribute `Content_JP`.

Advanced Object Mapping

By using adequately the `Parent` and `Sibling` attributes, Clarity is able to select an object in a dynamic list regardless of its position.

For such, the object and the sibling object should have the same parent object definition. However, the parent object definition can lead to a multiple object lists that will be filtered to a unique object containing the sibling object.

In the following example, in order to read the status associated to the product with the SKU "ANA079_NUDE", Clarity needs to find the table row of the product. In order to accomplish this, Clarity will look for an object `TableRow` which contains an object `SKUName` with the content: "ANA079_NUDE". The next step for Clarity is simply to look in that same row for the value of object: `Status`

Object	Object Definition
TableRow	Name: TableRow Type: css Value: tr
SKUName	Name: SKUName Type: css Value: td Content: ANA079_NUDE Parent: TableRow
Status	Name: Status Type: css Value: td Index: 10 Parent: TableRow Sibling: SKUName

Object: TableRow
The table row is the common parent for both the object and its sibling

	SKU	Price	Weight	Stock	Status	Created At
<input type="checkbox"/> 16497	Standard Anastasia Beverly Hills Matte Lipstick Soft Touch ANA079_TOUCH	£18.00	26g	10013	Ready for Live	22/05/2018
<input type="checkbox"/> 16497	Standard Anastasia Beverly Hills Matte Lipstick Nude ANA079_NUDE	£18.00	26g	27	Live	22/05/2018
<input type="checkbox"/> 16496	Standard Anastasia Beverly Hills Matte Lipstick Soft Touch ANA079_PETAL	£18.00	26g	3	Ready for Live	22/05/2018
<input type="checkbox"/> 16495	Standard Anastasia Beverly Hills Matte Lipstick Soft Touch ANA079_SWEET	£18.00	26g	0	Ready for Live	22/05/2018
<input type="checkbox"/> 16494	Standard Anastasia Beverly Hills Matte Lipstick Soft Touch ANA079_SEDON	£18.00	26g	0	Ready for Live	22/05/2018
<input type="checkbox"/> 16493	Standard Anastasia Beverly Hills Matte Lipstick Hollywood ANA079_HLLYWD	£18.00	26g	7	Ready for Live	22/05/2018
<input type="checkbox"/> 14426	Standard Anastasia Beverly Hills Matte Lipstick Orchid ANA079 Orchid	£18.00	24g	10008	Ready for Live	27/11/2017

Object: SKUName
Clarity uses the sibling object definition as a way to select the correct parent

Object: Status
The object is selected in the filtered parent

Requirement

The Requirement file is a YAML file containing a list of requirement information. This information is used to map the test scenarios and therefore compute the requirement test coverage.

The attributes to define the requirement are the following:

Attribute	Values	Description
Name*		Defines the name of the requirement The name should only contain numbers, letters, "-", "_" and ".".
Description		Defines the description of the requirement
System*		Defines the name of the system associated with the requirement
Origin		Defines the reference to a release at the origin of the requirement.
Domain		Defines the name of the domain associated with the requirement
Impact*	Integer > 0 and < 5	Defines the impact of failure associated with the requirement.
UsageFrequency*	Integer > 0 and < 5	Defines the frequency of usage of the requirement

* Mandatory attribute

Session Setup

The Session Setup file is a YAML file containing a list of session setup definitions. A session setup corresponds to a list of steps that will be executed before executing each test scenario.

The attributes to define the session setup are the following:

Attribute	Values	Description
Name*		Defines a name for the session setup. The name should only contain numbers, letters, "-", "_" and ".".
Steps*		Defines of list of steps associated with session setup. A step can be either a standard keyword, a business keyword or a custom keyword.

* Mandatory attribute

Test Scenario

The test scenario is a generic text file containing the list of steps (actions and validations) to be executed. Those steps are organized in sections delimited by “[” and “]”.

Section	Description
[Scenario]	Defines a set of generic attributes related to the test scenario.
[BeforeRun]	Defines the list of steps to execute before the [Content] section is executed.
[BeforeRunData]	Defines the data associated with the [BeforeRun] section.
[Content]	Defines the list of steps associated with the test scenario. This section is only executed if the [BeforeRun] section is executed successfully.
[Data]	Defines the data associated with the [Content] section.
[AfterRun]	Defines the list of steps to execute after the [Content] section is executed.
[AfterRunData]	Defines the data associated with the [AfterRun] section.
AfterRunOnSuccess]	Defines the list of steps to execute after the [Content] section is executed with success.
[AfterRunOnSuccessData]	Defines the data associated with the [AfterRunOnSuccess] section.
[AfterRunOnFailure]	Defines the list of steps to execute after the [Content] section is executed with a failure.
[AfterRunOnFailureData]	Defines the data associated with the [AfterRunOnFailure] section.
[AfterRunOnWarning]	Defines the list of steps to execute after the [Content] section is executed with a warning.
[AfterRunOnWarningData]	Defines the data associated with the [AfterRunOnWarning] section.
[AfterRunOnError]	Defines the list of steps to execute after the [Content] section is executed with an error.
[AfterRunOnErrorData]	Defines the data associated with the [AfterRunOnError] section.

Each section is executed for each data row defined in the respective data section. If the data section is empty, the corresponding section will be executed only once.

Test Results

The test execution results can have the following values:

Value	Description
Passed	Occurs when all test scenario steps are executed successfully.
Failed	Occurs when at least one test scenario step fails to execute (action) or when the expected value is different from the actual value (validation).
Failed Due To Dependency	Occurs as the consequence of having a dependency tests with a Failed result.
Warning	Occurs when at least a test scenario step generates a warning result and all other steps are executed successfully.
Error	Occurs when at least one test scenario step generates an unexpected result.

[Scenario] section

The section [Scenario] contains a set of attributes that defines the test scenario. The attributes should be inserted using the following syntax:

- Attribute name = Attribute value

Attribute	Values	Description
Name*		Defines the name of the test scenario. The name should only contain numbers, letters, "-", "_" and ".".
Description		Defines a description associated with the test scenario.
Author		Defines the author of the test scenario.
Requirement		Defines a list of requirements associated with the test scenario.
Defect		Defines the main defect associated with the test scenario.
ExecutionGroup		Defines the execution group associated with the test scenario. All the tests in the same execution group are executed sequentially.
AcceptanceCriteria		Defines the acceptance criteria associated with the test scenario.
Tags		Defines a list of tags used to categorize the test scenario.
Dependencies		Defines a logical expression used to identify the dependency with other test scenarios. Test scenarios are defined using regular expressions. Examples: <ul style="list-style-type: none">• Dependencies = (test01 or test02) and test03• Dependencies = test0[345] and test_API
DependencyDelay	Integer > 0	Defines the minimum elapse time between dependent test scenarios. The DependencyDelay value is expressed in seconds.
RunIterationSequentially	True False	Defines if the iterations of the content section of the test scenario needs to be executed sequentially.

* Mandatory attribute

Test Steps

The test steps are based on pre-defined keyword that are interpreted by the framework. Those keywords are classified in 3 categories:

Value	Description
Standard Keyword	Those keywords are defined as part of the framework core features and cannot be modified. The list of Standard keywords is defined in the Appendix A.
Business Keyword	A business keyword is a way to group standard keywords in order to perform a business operation.
Custom Keyword	A custom keyword is a way to extend the framework by adding bespoke keywords using custom Java libraries. The process of creation of custom keywords is described in Appendix K.

Test Data

Test data is divided in multiple types:

- Step Data
- Section Data

Step Data

Step data is the data needed by the test step to be executed. The data can be an input data, a validation data, an object name, a cookie name, etc...

Step data can also be defined using several approaches:

Step Data	Description
Hardcoded	The step data would be what the tester as set as part of the test step
From Javascript	The step data is defined as the execution of a JavaScript expression using the delimiters "\$(" and ")". Ex: \$(Date.now())
From Data	The step data is defined as a reference to data using the "{" and "}" characters. Ex: {Data1}
From Section Data	The step data is defined as a reference to a section data column using the "<" and ">" characters. Ex: <ColumnA>

For information, a data can be a reference to a section data value, and vice versa. For instance, both of those step data definition are valid:

- {Data1} where Data1 content is equal <ColumnA>
- <ColumnA> where columnA value is equal to {Data1}

Section Data

Section Data is data defined in complement to each section and used to iterate the section steps using different data configuration. The first row of the section data is used a column header to reference the section data as step data. For each following row, all the section steps would be executed.

In order to define the columns, the section data uses the “|” character to separate the values. As an example, the following definition contains 2 records of data. This configuration would lead to 2 iterations of the section steps. In order to use the data as part of step data, we can use respectively for the first and second column: <Name> and <City>. At each iteration, the reference to the section data will be replaced by the corresponding data. As an example, <Name> will be replaced by “John Smith” in the first iteration and “David Jones” for the second iteration.

Example:

```
[Data]
| Name      | City      |
| John Smith| New York  |
| David Jones| London   |
```

The section data can also be associated with a data of type Table. In that case, the section data should contain ONLY the reference to the data. When using a data, the first row of the data data will be considered as the headers for the data section.

Note: If the data does not contain any data, the section will be skipped during the execution.

Example:

```
[Data]
{Employees}
```

Test Exception

The Test Exception file is a YAML file containing a list of test exception definitions. A test exception corresponds to a list of steps that will be executed regularly in the background in order to deal with unplanned situations (ex: popup forms, etc...) that could interfere with the test execution process.

The attributes to define the test exception are the following:

Attribute	Values	Description
Name*		Defines a name for the test exception. The name should only contain numbers, letters, "-", "_" and ".".
StopOnSuccess	True False (Default)	Defines if the test exception should be interrupted if all steps are successful.
Steps*		Defines of list of steps associated with the test exception. A step can be either a standard keyword, a business keyword or a custom keyword.

* Mandatory attribute

Data

User-defined Data

The Data file is a YAML file containing a list of data definitions. Defining a data allows to initialize parameters used during the test execution.

The attributes to define the data are the following:

Attribute	Values	Description
Name*		Defines a name for the data. The name should only contain numbers, letters, "-" and "_"
Type*	String Table TextFile BinaryFile URL Folder	Defines the type of the data.
Value*		Defines the value of the data of type: String. This attribute is mandatory if the attribute MultipleValue is not defined.
MultipleValue*		Defines the value of the attribute to use as the data value. This attribute is only available for data of type String. This attribute is mandatory if the attribute Value is not defined.
Path*		Defines the path of the file associated with the types: Table, Folder, TextFile or BinaryFile.
URL*		Defines the url associated with the type: URL.
Persistent	False (Default) True	Defines if the file associated with a data of type Table is saved on disk when the content of the data is updated.

* Mandatory attribute

Local and Global Data

- Local data are data restricted to the duration of a test execution.
- Global data are data available during the duration of the configuration execution. As a consequence, global data would allow the exchange of data between tests.

Data Usage

To access the content of a data in a test, the data name should be used between the characters: "{" and "}". As an example, "{VarProductName}" will return the value of data "VarProductName"

If no local data is found, Clarity will try to find a global data with the same name. In order to force the usage of a global data, the data name should be prefixed with "*". Therefore, "{*ProductID}" would return the content of the global data "ProductID".

MultipleValue

In order to associate multiple value to a data, the attribute `MultipleValue` can be used. In this case, `MultipleValue` contains the name of the attribute to be used as the data value.

For example, in a Multilanguage system, the content of a data can contain a value translated in each language. In order to select the right value based on the country the following data definition can be used:

```
- Name: CreateAccount
  Type: String
  MultipleValue: Value_{CountryCode}
  Value_US: "Create an Account"
  Value_JP: "アカウントを登録する"
```

If the data `CountryCode` contains the value "US", the attribute `Value_US` will be used as the value of data `CreateAccount`. If the `CountryCode` is equal to "JP", the framework will use instead the attribute `Value_JP`.

Technical Data

In complement to user-defined data, the framework includes a set of technical data

Name	Description
<code>\$UUID</code>	This data contains a unique ID. The unique ID has the format of a Universal Unique ID (ex: 123e4567-e89b-12d3-a456-426655440000) that can be used to make the test data unique during execution (ex: <code>TestData_{UUID}</code>)
<code>\$Return</code>	This data contains a return character
<code>\$TimeStampID</code>	This data contains a unique ID. The unique ID represents the number of milliseconds since: January 1, 1970, 00:00:00 GMT
<code>\$TestName</code>	This data contains the test scenario name.
<code>\$DataIteration</code>	This data contains the section data iteration number.
<code>\$QA</code>	This data contains a string only displayed when the QA command option is used.
<code>\$User.Firstname</code>	This data contains the first name of the Clarity user
<code>\$User.Lastname</code>	This data contains the last name of the Clarity user
<code>\$User.Name</code>	This data contains the name of the Clarity user
<code>\$User.Email</code>	This data contains the email of the Clarity user
<code>\$Env. + Environment data name</code>	This data contains the value of the environment data
<code>\$\$ + Data Pattern</code>	This data contains a data generated using the defined data pattern
<code>\$LoopObject</code>	This data contains the current object associated with a <code>@foreachobject</code> loop

Technical Data for Capabilities

Name	Description
\$Capability.Name	This data contains the name of the capability used for the test execution.
\$Capability.Type	This data contains the type of the capability used for the test execution.
\$Capability.IsBrowser	This data contains the value "True" if the capability used for the test execution corresponds to one of the following configurations: <ul style="list-style-type: none">Type = BrowserType = BrowserStack and DeviceType=Browser. Otherwise, the return value is "False"
\$Capability.Browser.Name	This data contains the name of the browser used for the test execution

Data Type: TextFile

Data of type TextFile are data associated with a text file and which value.

By default, data of type TextFile are defined as Global data.

Data Type: BinaryFile

Data of type BinaryFile are data associated with a Binary file (ex. Images, videos, ...).

By default, data of type BinaryFile are defined as Global data.

Data Type: URL

Data of type URL are data associated with an URL of a Binary file.

By default, data of type URL are defined as Global data.

Data Type: Table

Data of type Table are 2-dimensions data defined by columns and rows.

The first row contains the list of the column names. The column name should follow the same naming rules as used for data and should contain a single column with the name "ID".

The values contain in the column "ID" should be unique and used to identify the table row by the selection keywords.

By default, data of type Table are defined as Global data.

Data Table: Selection

The table selection keywords are:

- SelectTableWithOption
- SelectTableWhereColumnEqualTo
- SelectTableWhereColumnEqualToWithOption
- SelectTableWhereColumnNotEqualTo
- SelectTableWhereColumnNotEqualToWithOption

Those keywords allow to select a row in a table data. When a row is selected, the framework generates a list of data corresponding to the column values. Those data are defined as data of type String and used the following naming convention: "Table data name"."column name".

Instead of using keywords, the selection of a data of type Table can also be done using the following syntaxes:

- {data # option}
- {data @ column = value }
- {data @ column = value # option }
- {data @ column != value }
- {data @ column != value # option }

The list of possible options is:

Data Option	Description
Sequential (Default)	Select sequentially a value from the data or the result of the filter.
Random	Select randomly a value from the data or the result of the filter.
Available-Sequential Available	Select sequentially a value from the list of value of the data (or from the result of the filter) not being used at the time of the value request.
Available-Random	Select randomly a value from the list of value of the data (or from the result of the filter) not being used at the time of the value request.
Unique-Sequential Unique	Select sequentially a value that has never been used from the list of value of the data or from the result of the filter
Unique-Random	Select randomly a value that has never been used from the list of value of the data or from the result of the filter.

Data Pattern

The framework can generate or validate data using a data pattern as an input. To generate data, the data pattern should be used in a technical data. In order to use a data pattern for a validation, a specific validation keyword accepting data pattern should be used.

A data pattern is defined by a comma separated list of data expressions: expression1, expression2, ..., etc

Data expressions can be chosen from the following list:

Data Expression	Description	Expression Example	Generation Example
!	character !	!	!
!text	text	!N5	N5
Non data expression text	Text not matching a data pattern expression	Abc	abc
N	random number	N	15
N + [format]	random number formatted	N [0.00]	1023.99
N + numeric value	random string of n digits numbers	N3	264
n + numeric value	random string of n binary numbers	n4	0110
S + numeric value	random string of n letters	S5	GafEr
S + numeric value + [U]	random string of n uppercase letters	S4 [U]	AKEX
S + numeric value + [L]	random string of n lowercase letters	s2 [L]	qp
X + numeric value	random string of n letters and numbers	X5	V2Qol
X + numeric value + [U]	random string of n uppercase letters and numbers	X4 [U]	R8OC
X + numeric value + [L]	random string of n lowercase letters and numbers	X2 [L]	4m

D	<i>current date</i>	D	2016-02-05
D + date + [date format]	<i>formatted date</i>	D2016-02-05 [dd/MM/YYYY]	05/02/2016
T	<i>current time</i>	T	15:10:45
T + time + [time format]	<i>formatted time</i>	T14:17:23 [HH:mm]	14:17
DT	<i>current date time</i>	DT	2016-01-28 14:18:15.324
DT + datetime + [datetime format]	<i>formatted date+time</i>	DT2016-02-05 14:18:15 [HH:mm:ss dd/MM/YYYY]	14:18:15 05/02/2016
B	<i>random boolean value</i>	B	TRUE
TRUE	<i>Boolean true value</i>	TRUE	TRUE
FALSE	<i>Boolean false value</i>	FALSE	FALSE
IN[min value max value][format]	<i>numeric value within an interval</i>	IN[10 15] [0.00]	11.56
ID[min value max value][format]	<i>date value within an interval</i>	ID[2016-01-01 2016-01-31] [dd/MM/YYYY]	15/01/2016
IT[min value max value][format]	<i>time value within an interval</i>	IT[10:00:00 11:00:00] [HH:mm]	10:29
IDT[min value max value][format]	<i>date time value within an interval</i>	IDT[2016-01-01 00:00:00 2016-01-31 23:59:59] [dd/MM/YYYY HH:mm]	15/01/2016 12:34

Default Format

Data Type	Format
Number	0
Date	yyyy-MM-dd
Time	HH:mm:ss
DateTime	yyyy-MM-dd HH:mm:ss

Data Pattern Examples

Example	Format
S10, @, !server.com	YjgRFbKOWS@server.com
!SKU, N4	SKU5389

Appendix A – Functional Keywords

Actions

Category	Name	Description	Example
Browser	RestartBrowser	Restarts the browser	The user restarts the browser
Browser	NavigateTo	Navigates to (data)	The user navigates to http://google.com
Browser	NavigateBack	Navigates Back	The user navigates back
Browser	NavigateForward	Navigates forward	The user navigates forward
Browser	NavigateToDefault	Navigates to the default URL	The user navigates to The default url
Browser	PageRefresh	Performs a page refresh	The user performs a page refresh
Browser	ScrollBottomPage	scrolls to the bottom of the page	The user scrolls to The bottom of The page
Browser	ScrollTopPage	scrolls to the top of the page	The user scrolls to The top of The page
Browser	OpenNewTab	opens new tab	The user opens new tab
Browser	CloseTabName	closes tab name (data)	The user closes tab name cnn
Browser	CloseTab	closes tab (data)	The user closes tab {TabCNN}
Browser	SelectTabName	selects tab name (data)	The user selects tab name cnn
Browser	SelectTab	selects tab (data)	The user selects tab {TabCNN}
Browser	ResetTabs	resets the tabs	The user resets the tabs
Browser	SetHeader	sets Header (attribute) with value (data)	The user sets Header SessionID with value ABC123
Browser	DeleteHeader	Deletes Header (attribute)	The user deletes Header SessionID
Browser	DeleteAllHeaders	Deletes All Headers	The user deletes all Headers
Category	Name	Description	Example
Session	StartBrowserSession	starts browser session (data)	The user starts browser session session2
Session	CloseBrowserSession	closes browser session (data)	The user closes browser session session2
Session	UseDefaultBrowserSession	uses default browser session	The user uses default browser session
Session	UseBrowserSession	uses browser session (data)	The user uses browser session session2
Session	ResetSessions	resets browser sessions	The user resets browser sessions
Category	Name	Description	Example
Object	SetValueInObject	Sets the value (data) in object (object)	The user sets The value "abc" in object username
Object	SetClipboardInObject	Sets clipboard in object (object)	The user sets clipboard in object username
Object	ClearObject	Clears object (object)	The user clears object username
Object	TypeValueInObject	Types the value (data) in object (object)	The user types The value "abc" in object username
Object	SelectValueAtIndexInObject	Selects the value at index (data) in object (object)	The user selects the value at index 1 in object options
Object	SelectValueInObject	Selects the value (data) in object (object)	The user selects the value "option 1" in object options
Object	DragAndDrop	Drag and drop object (object) into object (object)	The user drag and drop object DragImage1 into object DragAndDropTarget1
Object	DoubleClick	Doubleclicks on object (object)	The user doubleclicks on object buttonOK
Object	Click	Clicks on object (object)	The user clicks on object buttonOK
Object	Uncheck	Unchecks object (object)	The user unchecks object checkBox
Object	Check	Checks object (object)	The user checks object checkBox
Object	HoverOver	Hovers over object (object)	The user hovers over object serviceLink
Object	ScrollToObject	Scrolls to object (object)	The user scrolls to object serviceLink
Object	SaveObjectFromObjectMapping	Save object (objectname) from object mapping (object)	The user saves object productXYZ from object mapping product
Object	UploadFile	uploads file (data) using object (object)	The user uploads file image1 using object uploadObject
Object	UploadURL	uploads url (data) using object (object)	The user uploads url (data) using object (object)
Category	Name	Description	Example
Timer	WaitForSecond	Waits for (data) seconds	The user waits ffor 5 seconds
Category	Name	Description	Example
Data	SetDataWithObjectValue	Sets data (data) with value of object (object)	the user sets data data1 with value of object username
Data	SetDataWithObjectContent	Sets data (data) with content of object (object)	the user sets data data1 with content of object username
Data	SetDataWithObjectSelection	Sets data (data) with selection of object (object)	the user sets data data1 with selection of object username
Data	SetDataWithObjectSelectionIndex	Sets data (data) with selection index of object (object)	the user sets data data1 with selection index of object username
Data	SetDataWithObjectHTML	Sets data (data) with HTML content of object (object)	the user sets data data1 with HTML content of object username
Data	SetDataWithObjectCount	Sets data (data) with object count of (object)	the user sets data data1 with object count of products
Data	SetDataWithValue	Sets data (data) with value (data)	The user sets data data1 with value "abc"
Data	SetDataWithURL	Sets data (data) with current url	The user sets data data1 with current url
Data	SetDataWithClipboard	Sets data (data) with clipboard	The user sets data data1 with clipboard
Data	SetDataWithCookieValue	Sets data (data) with value of cookie (cookie)	The user sets data data1 with value of cookie sessionID
Data	SetDataWithCookieDomain	Sets data (data) with domain of cookie (cookie)	The user sets data data1 with domain of cookie sessionID
Data	SetDataWithCookiePath	Sets data (data) with path of cookie (cookie)	The user sets data data1 with path of cookie sessionID
Data	SetDataWithCookieExpiryDate	Sets data (data) with expiry date of cookie (cookie)	The user sets data data1 with expiry date of cookie sessionID
Data	SetDataWithPageTitle	Sets data (data) with page title	The user sets data data1 with page title
Data	SetDataWithPageSource	Sets data (data) with page source	The user sets data data1 with page source
Data	SetDataWithScriptResultInBrowser	Sets data (data) with result of (data) in browser	The user sets data data1 with result of "return document.readyState.toLowerCase()" in browser
Data	SetDataWithURLReturnCode	Sets data (data) with return code of URL (data)	The system sets data ReturnCode with return code of URL "http://www.google.com"

Data	SaveDataInFile	Saves data (data) in folder (data) with the file name (data)	The user saves data record in folder "target" with the file name "result.txt"
Data	SetDataWithCommandResult	Sets data (data) with result of command (data)	The user sets data listFile with result of command dir *txt
Category	Name	Description	Example
Data	DeleteGlobalData	Deletes global data (data)	the user deletes global data (data)
Data	SetGlobalDataWithObjectValue	Sets global data (data) with value of object (object)	the user sets global data data1 with value of object username
Data	SetGlobalDataWithObjectContent	Sets global data (data) with content of object (object)	the user sets global data data1 with content of object username
Data	SetGlobalDataWithObjectSelection	Sets global data (data) with selection of object (object)	the user sets global data data1 with selection of object username
Data	SetGlobalDataWithObjectSelectionIndex	Sets global data (data) with selection index of object (object)	the user sets global data data1 with selection index of object username
Data	SetGlobalDataWithObjectHTML	Sets global data (data) with HTML content of object (object)	the user sets global data data1 with HTML content of object username
Data	SetGlobalDataWithObjectCount	Sets global data (data) with object count of (object)	the user sets global data data1 with object count of products
Data	SetGlobalDataWithValue	Sets global data (data) with value (data)	The user sets global data data1 with value "abc"
Data	SetGlobalDataWithURL	Sets global data (data) with current url	The user sets global data data1 with current url
Data	SetGlobalDataWithClipboard	Sets global data (data) with clipboard	The user sets global data data1 with clipboard
Data	SetGlobalDataWithCookieValue	Sets global data (data) with value of cookie (cookie)	The user sets global data data1 with value of cookie sessionID
Data	SetGlobalDataWithCookieDomain	Sets global data (data) with domain of cookie (cookie)	The user sets global data data1 with domain of cookie sessionID
Data	SetGlobalDataWithCookiePath	Sets global data (data) with path of cookie (cookie)	The user sets global data data1 with path of cookie sessionID
Data	SetGlobalDataWithCookieExpiryDate	Sets global data (data) with expiry date of cookie (cookie)	The user sets global data data1 with expiry date of cookie sessionID
Data	SetGlobalDataWithPageTitle	Sets global data (data) with page title	The user sets global data data1 with page title
Data	SetGlobalDataWithPageSource	Sets global data (data) with page source	The user sets global data data1 with page source
Data	SetGlobalDataWithScriptResultInBrowser	Sets global data (data) with result of (script) in browser	The user sets global data data1 with result of script1 in browser
Data	SetGlobalDataWithURLReturnCode	Sets global data (data) with return code of URL (data)	The system sets global data ReturnCode with return code of URL "http://www.google.com"
Data	SaveGlobalDataInFile	Saves global data (data) in folder (data) with the file name (data)	The user saves global data record in folder "target" with the file name "result.txt"
Data	SetGlobalDataWithCommandResult	Sets global data (data) with result of command (data)	the user sets global data listFile with result of command dir *txt
Category	Name	Description	Example
Data-Browser	SetDataWithTabTitle	Sets data (data) with current tab title	The user sets data data1 with current tab title
Data-Browser	SetDataWithTab	Sets data (data) with current tab	The user sets data data1 with current tab
Data-Browser	SetGlobalDataWithTabTitle	Sets global data (data) with current tab title	The user sets global data data1 with current tab title
Data-Browser	SetGlobalDataWithTab	Sets global data (data) with current tab	The user sets global data data1 with current tab
Category	Name	Description	Example
Data-File	SetDataAsTextFile	Sets data (data) from text file name (data) in folder (data)	Sets data varFile from text file name "result.txt" in folder "target"
Data-File	SetDataAsBinaryFile	Sets data (data) from binary file name (data) in folder (data)	Sets data varFile from binary file name "result.txt" in folder "target"
Data-File	UpdateDataFromFile	Updates data (data) from file	Updates data varFile from file
Data-File	SetGlobalDataAsTextFile	Sets global data (data) from text file name (data) in folder (data)	Sets global data varFile from text file name "result.txt" in folder "target"
Data-File	SetGlobalDataAsBinaryFile	Sets global data (data) from binary file name (data) in folder (data)	Sets global data varFile from binary file name "result.txt" in folder "target"
Data-File	UpdateGlobalDataFromFile	Updates global data (data) from file	Updates global data varFile from file
Category	Name	Description	Example
Table Data	SetDataWithTableRow	Sets data (data) with table row (data)	The user sets data data1 with table row "Product1"
Table Data	SetGlobalDataWithTableRow	Sets global data (data) with table row (data)	The user sets global data data1 with table row "Product1"
Table Data	SelectGlobalTableWhereColumnEqualToWithOption	Selects global table (data) where column id is equal to (data) with option (data)	The user selects record in global table Country where column ID is equal to UK with option available-sequential
Table Data	SelectTableWhereColumnEqualToWithOption	Selects record in table (data) where column id is equal to (data) with option (data)	The user selects record in table Country where column ID is equal to UK with option available-sequential
Table Data	SelectGlobalTableWhereColumnEqualTo	Selects record in global table (data) where column id is equal to (data)	The user selects record in global table Country where column ID is equal to UK
Table Data	SelectTableWhereColumnEqualTo	Selects record in table (data) where column id is equal to (data)	The user selects record in table Country where column ID is equal to UK
Table Data	SelectGlobalTableWhereColumnNotEqualToWithOption	Selects record in global table (data) where column id is not equal to (data) with option (data)	The user selects record in global table Country where column ID is not equal to UK with option available-sequential
Table Data	SelectTableWhereColumnNotEqualToWithOption	Selects record in table (data) where column id is not equal to (data) with option (data)	The user selects record in table Country where column ID is not equal to UK with option available-sequential
Table Data	SelectGlobalTableWhereColumnNotEqualTo	Selects record in global table (data) where column id is not equal to (data)	The user selects record in global table Country where column ID is not equal to UK
Table Data	SelectTableWhereColumnNotEqualTo	Selects record in table (data) where column id is not equal to (data)	The user selects record in table Country where column ID is not equal to UK

Table Data	SelectGlobalTableWithOption	Selects record in global table (data) with option (data)	The user selects record in global table ExecGroup with option available-sequential
Table Data	SelectTableWithOption	Selects record in table (data) with option (data)	The user selects record in table ExecGroup with option available-sequential
Table Data	SelectGlobalTable	Selects record in global table (data)	The user selects record in global table ExecGroup
Table Data	SelectTable	Selects record in table (data)	The user selects record in table ExecGroup
Table Data	ResetSequenceTable	Resets sequence of table (data)	The user resets sequence of table ExecGroup
Table Data	ResetSequenceGlobalTable	Resets sequence of global table (data)	The user resets sequence of global table ExecGroup
Table Data	ReleaseTableRow	Releases table row (data) on table (data)	The user releases table row CountryUK on table Country
Table Data	SetCellInTableRowWithValue	Sets cell (data) in table row (data) with value (data)	The user sets cell ID in table row Product1 with value Product
Table Data	SaveTableRowInTable	Saves table row (data) in table (data)	The user saves table row ProductRow in table Product
Table Data	SaveTableRowInGlobalTable	Saves table row (data) in global table (data)	The user saves table row ProductRow in global table Product
Table Data	SaveGlobalTableRowInTable	Saves global table row (data) in table (data)	The user saves global table row ProductRow in table Product
Table Data	SaveGlobalTableRowInGlobalTable	Saves global table row (data) in global table (data)	The user saves global table row ProductRow in global table Product
Table Data	DeleteTableRowInTable	Deletes table row (data) in table (data)	The user deletes table row ProductRow in table Product
Table Data	DeleteTableRowInGlobalTable	Deletes table row (data) in global table (data)	The user deletes table row ProductRow in global table Product
Table Data	CreateTableRowDataFromTable	Creates table row data (data) with columns from table (data)	The user creates table row data Country with columns from table GlobalCountry
Table Data	CreateTableRowData	Creates table row data (data) with columns (data)	The user creates table row data Country with columns [ID, Name]
Table Data	CreateTableRowGlobalDataFromTable	Creates table row global data (data) with columns from table (data)	The user creates table row global data Country with columns from table GlobalCountry
Table Data	CreateTableRowGlobalData	Creates table row global data (data) with columns (data)	The user creates table row global data Country with columns [ID, Name]
Table Data	CreateTableDataFromTable	Creates table data (data) with columns from table (data)	The user creates table data Country with columns from table GlobalCountry
Table Data	CreateTableData	Creates table data (data) with columns (data)	The user creates table data Country with columns [ID, Name]
Table Data	CreateTableGlobalDataFromTable	Creates table global data (data) with columns from table (data)	The user creates table global data Country with columns from table GlobalCountry
Table Data	CreateTableGlobalData	Creates table global data (data) with columns (data)	The user creates table global data Country with columns [ID, Name]
Category	Name	Description	Example
Cookie	SetCookieForDomainAndPath	Set Cookie (cookie) with value (data1) for domain (data2) and path (data3)	The user sets Cookie SessionID with value abc123 for domain google.com and path "/"
Cookie	DeleteCookie	Deletes cookie (data)	The user deletes cookie sessionID
Cookie	DeleteAllCookies	Deletes all cookies	The user deletes all cookies
Category	Name	Description	Example
JavaScript	ExecuteScriptInBrowser	Executes script (script) in the browser	The system executes script "return document.readyState.toLowerCase()" in the browser
Category	Name	Description	Example
File	SaveTextInFile	Saves value (data) in folder (data) with the file name (data)	The user saves value "text value" in folder "target" with the file name "result.txt"
File	SaveTextInDataFile	Saves value (data) in text file data (data)	The user saves value "text" in text file data textfile
File	SaveTextInGlobalDataFile	Saves value (data) in text file global data (data)	The user saves value "text" in text file global data textfile
File	DeleteFile	Deletes file name (data) in folder (data)	Deletes file name "result.txt" in folder "target"
File	DeleteDataFile	Deletes file of data (data)	Deletes file of data varFile
File	DeleteGlobalDataFile	Deletes file of global data (data)	Deletes file of global data varFile
Category	Name	Description	Example
System	AddReportLabel	Adds report label: (data)	The user adds report label: Config 1
Console	SendToConsole	Sends value (data) to console	The user sends value "text" to console
System	ExecuteCommand	Executes command (data)	The system executes command dir
Category	Name	Description	Example
Clipboard	SetClipboardWithObjectValue	Sets clipboard with value of object (object)	the user sets clipboard with value of object username
Clipboard	SetClipboardWithObjectContent	Sets clipboard with content of object (object)	the user sets clipboard with content of object username
Clipboard	SetClipboardWithObjectSelection	Sets clipboard with selection of object (object)	the user sets clipboard with selection of object username
Clipboard	SetClipboardWithObjectSelectionIndex	Sets clipboard with selection index of object (object)	the user sets clipboard with selection index of object username
Clipboard	SetClipboardWithObjectHTML	Sets clipboard with HTML content of object (object)	the user sets clipboard with HTML content of object username
Clipboard	SetClipboardWithObjectCount	Sets clipboard with object count of (object)	the user sets clipboard with object count of products
Clipboard	SetClipboardWithURL	Sets clipboard with current url	The user sets clipboard with current url
Clipboard	SetClipboardWithCookieValue	Sets clipboard with value of cookie (cookie)	The user sets clipboard with value of cookie sessionID
Clipboard	SetClipboardWithCookieDomain	Sets clipboard with domain of cookie (cookie)	The user sets clipboard with domain of cookie sessionID
Clipboard	SetClipboardWithCookiePath	Sets clipboard with path of cookie (cookie)	The user sets clipboard with path of cookie sessionID
Clipboard	SetClipboardWithCookieExpiryDate	Sets clipboard with expiry date of cookie (cookie)	The user sets clipboard with expiry date of cookie sessionID
Clipboard	SetClipboardWithPageTitle	Sets clipboard with page title	The user sets clipboard with page title
Clipboard	SetClipboardWithPageSource	Sets clipboard with page source	The user sets clipboard with page source
Clipboard	SetClipboardWithScriptResultInBrowser	Sets clipboard with result of (data) in browser	The user sets clipboard with result of "return document.readyState.toLowerCase()" in browser

Clipboard	SetClipboardWithURLReturnCode	Sets clipboard with return code of URL (data)	The system sets clipboard with return code of URL "http://www.google.com"
Clipboard	SetClipboardWithCommandResult	Sets clipboard with result of command (data)	The user sets clipboard with result of command dir *txt
Clipboard	SetClipboardWithValue	Sets clipboard with value (data)	The user sets clipboard with value "abc"

Validations

Category	Name	Description	Example
Browser	UrlEqualTo	Current url should be equal to (data)	The current url should be equal to http://google.com
Browser	UrlNotEqualTo	Current url should be not equal to (data)	The current url should be not equal to http://google.com
Browser	UrlRegexMatch	Current url should match (data)	The current url should match http://google.com
Browser	UrlRegexNotMatch	Current url should not match (data)	The current url should not match http://google.com
Category	Name	Description	Example
Object	CountEqualTo	The count of object (object) should be equal to (data)	The count of object button should be equal to 1
Object	CountNotEqualTo	The count of object (object) should be not equal to (data)	The count of object button should be not equal to 1
Object	CountGreaterThan	The count of object (object) should be greater than (data)	The count of object button should be greater than 1
Object	CountGreaterThanOrEqual	The count of object (object) should be greater or equal to (data)	The count of object button should be greater or equal to 1
Object	CountLowerThan	The count of object (object) should be lower than (data)	The count of object button should be lower than 1
Object	CountLowerThanOrEqual	The count of object (object) should be lower or equal to (data)	The count of object button should be lower or equal to 1
Object	ValueEqualTo	The value of object (object) should be equal to (data)	The value of object username should be equal to "abc"
Object	ValueNotEqualTo	The value of object (object) should be not equal to (data)	The value of object username should be not equal to "abc"
Object	ValueGreaterThan	The value of object (object) should be greater than (data)	The value of object quantity should be greater than 2
Object	ValueGreaterThanOrEqual	The value of object (object) should be greater or equal to (data)	The value of object quantity should be greater or equal to 2
Object	ValueLowerThan	The value of object (object) should be lower than (data)	The value of object quantity should be lower than 2
Object	ValueLowerThanOrEqual	The value of object (object) should be lower or equal to (data)	The value of object quantity should be lower or equal to 2
Object	ValueStartWith	The value of object (object) should start with (data)	The value of object username should start with "abc"
Object	ValueNotStartWith	The value of object (object) should not start with (data)	The value of object username should not start with "abc"
Object	ValueEndWith	The value of object (object) should end with (data)	The value of object username should end with "abc"
Object	ValueNotEndWith	The value of object (object) should not end with (data)	The value of object username should not end with "abc"
Object	ValueContains	The value of object (object) should contain (data)	The value of object username should contain "abc"
Object	ValueNotContain	The value of object (object) should not contain (data)	The value of object username should not contain "abc"
Object	ValuePatternMatch	The value of object (object) should match pattern (data)	The value of object username should match pattern "N2"
Object	ValuePatternNotMatch	The value of object (object) should not match pattern (data)	The value of object username should not match pattern "N2"
Object	ValueRegexMatch	The value of object (object) should match (data)	The value of object username should match "abc"
Object	ValueRegexNotMatch	The value of object (object) should not match (data)	The value of object username should not match "abc"
Object	HTMLContentEqualTo	The HTML content of object (object) should be equal to (data)	The HTML content of object username should be equal to "abc"
Object	HTMLContentNotEqualTo	The HTML content of object (object) should be not equal to (data)	The HTML content of object username should be not equal to "abc"
Object	HTMLContentStartWith	The HTML content of object (object) should start with (data)	The HTML content of object username should start with "abc"
Object	HTMLContentNotStartWith	The HTML content of object (object) should not start with (data)	The HTML content of object username should not start with "abc"
Object	HTMLContentEndWith	The HTML content of object (object) should end with (data)	The HTML content of object username should end with "abc"
Object	HTMLContentNotEndWith	The HTML content of object (object) should not end with (data)	The HTML content of object username should not end with "abc"
Object	HTMLContentContains	The HTML content of object (object) should contain (data)	The HTML content of object username should contain "abc"
Object	HTMLContentNotContain	The HTML content of object (object) should not contain (data)	The HTML content of object username should not contain "abc"
Object	HTMLContentRegexMatch	The HTML content of object (object) should match (data)	The HTML content of object username should match "abc"
Object	HTMLContentRegexNotMatch	The HTML content of object (object) should not match (data)	The HTML content of object username should not match "abc"
Object	SelectionIndexEqualTo	The selection index in object (object) should be equal to (data)	The selection index of object country should be equal to 1
Object	SelectionIndexNotEqualTo	The selection index in object (object) should be not equal to (data)	The selection index of object country should be not equal to 1
Object	SelectionEqualTo	The selection in object (object) should be equal to (data)	The selection of object country should be equal to "England"
Object	SelectionNotEqualTo	The selection in object (object) should be not equal to (data)	The selection of object country should be not equal to "England"

Object	ContentEqualTo	The content of object (object) should be equal to (data)	The content of object country should be equal to ["England", "Wales"]
Object	ContentNotEqualTo	The content of object (object) should be not equal to (data)	The content of object country should be not equal to ["England", "Wales"]
Object	ContentContains	The content of object (object) should contain (data)	The content of object country should contain ["England", "Wales"]
Object	ContentDoesNotContain	The content of object (object) should not contain (data)	The content of object country should not contain ["England", "Wales"]
Object	Selected	The object (object) should be selected	The object checkbox should be selected
Object	NotSelected	The object (object) should be not selected	The object checkbox should be not selected
Cookie	CookieEqualTo	Cookie (cookie) should be equal to (data)	The cookie sessionID should be equal to "abc"
Cookie	CookieNotEqualTo	Cookie (cookie) should be not equal to (data)	The cookie sessionID should be not equal to "abc"
Category	Name	Description	Example
JavaScript	ScriptResultInBrowserEqualTo	The result of script (script) in the browser should be equal to (data)	The result of script "return document.readyState.toLowerCase()" in the browser should be equal to "complete"
JavaScript	ScriptResultInBrowserNotEqualTo	The result of script (script) in the browser should be not equal to (data)	The result of script "return document.readyState.toLowerCase()" in the browser should be not equal to ""
Category	Name	Description	Example
Data	DataEqualTo	The value of data (data) should be equal to (data)	The value of data data1 should be equal to "abc"
Data	DataNotEqualTo	The value of data (data) should be not equal to (data)	The value of data data1 should be not equal to "abc"
Data	DataGreaterThan	The value of data (data) should be greater than (data)	The value of data data1 should be greater than 2
Data	DataGreaterThanOrEqual	The value of data (data) should be greater or equal to (data)	The value of data data1 should be greater or equal to 2
Data	DataLowerThan	The value of data (data) should be lower than (data)	The value of data data1 should be lower than 2
Data	DataLowerThanOrEqual	The value of data (data) should be lower or equal to (data)	The value of data data1 should be lower or equal to 2
Data	DataStartWith	The value of data (data) should start with (data)	The value of data data1 should start with "abc"
Data	DataDoesNotStartWith	The value of data (data) should not start with (data)	The value of data data1 should not start with "abc"
Data	DataEndWith	The value of data (data) should end with (data)	The value of data data1 should end with "abc"
Data	DataDoesNotEndWith	The value of data (data) should not end with (data)	The value of data data1 should not end with "abc"
Data	DataContains	The value of data (data) should contain (data)	The value of data data1 should contain "abc"
Data	DataDoesNotContain	The value of data (data) should not contain (data)	The value of data data1 should not contain "abc"
Data	DataPatternMatch	The value of data (data) should match pattern (data)	The value of data data1 should match pattern "N2"
Data	DataPatternNotMatch	The value of data (data) should not match pattern (data)	The value of data data1 should not match pattern "N2"
Data	DataRegExMatch	The value of data (data) should match (data)	The value of data data1 should match "abc"
Data	DataRegExNotMatch	The value of data (data) should not match (data)	The value of data data1 should not match "abc"
Category	Name	Description	Example
Data	GlobalDataEqualTo	The value of global data (data) should be equal to (data)	The value of global data data1 should be equal to "abc"
Data	GlobalDataNotEqualTo	The value of global data (data) should be not equal to (data)	The value of global ariable data1 should be not equal to "abc"
Data	GlobalDataGreaterThan	The value of global data (data) should be greater than (data)	The value of global data data1 should be greater than 2
Data	GlobalDataGreaterThanOrEqual	The value of global data (data) should be greater or equal to (data)	The value of global data data1 should be greater or equal to 2
Data	GlobalDataLowerThan	The value of global data (data) should be lower than (data)	The value of global data data1 should be lower than 2
Data	GlobalDataLowerThanOrEqual	The value of global data (data) should be lower or equal to (data)	The value of global data data1 should be lower or equal to 2
Data	GlobalDataStartWith	The value of global data (data) should start with (data)	The value of global data data1 should start with "abc"
Data	GlobalDataDoesNotStartWith	The value of global data (data) should not start with (data)	The value of global data data1 should not start with "abc"
Data	GlobalDataEndWith	The value of global data (data) should end with (data)	The value of global data data1 should end with "abc"
Data	GlobalDataDoesNotEndWith	The value of global data (data) should not end with (data)	The value of global data data1 should not end with "abc"
Data	GlobalDataContains	The value of global data (data) should contain (data)	The value of global data data1 should contain "abc"
Data	GlobalDataDoesNotContain	The value of global data (data) should not contain (data)	The value of global data data1 should not contain "abc"
Data	GlobalDataPatternMatch	The value of global data (data) should match pattern (data)	The value of global data data1 should match pattern "N2"
Data	GlobalDataPatternNotMatch	The value of global data (data) should not match pattern (data)	The value of global data data1 should not match pattern "N2"
Data	GlobalDataRegExMatch	The value of global data (data) should match (data)	The value of global data data1 should match "abc"
Data	GlobalDataRegExNotMatch	The value of global data (data) should not match (data)	The value of global data data1 should not match "abc"
Category	Name	Description	Example
Data-File	DataFileExist	The data file (filename) should exist	The data file (filename) should exist
Data-File	DataFileNotExist	The data file (filename) should not exist	The data file (filename) should not exist
Data-File	GlobalDataFileExist	The global data file (filename) should exist	The global data file (filename) should exist
Data-File	GlobalDataFileNotExist	The global data file (filename) should not exist	The global data file (filename) should not exist
Category	Name	Description	Example
File	FileExist	The file (filename) should exist	The file (filename) should exist
File	FileNotExist	The file (filename) should not exist	The file (filename) should not exist
Category	Name	Description	Example
System	CommandResultEqualTo	The result of command (data) should be equal to (data)	The result of command "type file.txt" should be equal to "abc"
System	CommandResultNotEqualTo	The result of command (data) should be not equal to (data)	The result of command "type file.txt" should be not equal to "abc"

System	CommandResultGreaterThan	The result of command (data) should be greater than (data)	The result of command "type file.txt" should be greater than 2
System	CommandResultGreaterThanOrEqual	The result of command (data) should be greater or equal to (data)	The result of command "type file.txt" should be greater or equal to 2
System	CommandResultLowerThan	The result of command (data) should be lower than (data)	The result of command "type file.txt" should be lower than 2
System	CommandResultLowerThanOrEqual	The result of command (data) should be lower or equal to (data)	The result of command "type file.txt" should be lower or equal to 2
System	CommandResultStartWith	The result of command (data) should start with (data)	The result of command "type file.txt" should start with "abc"
System	CommandResultDoesNotStartWith	The result of command (data) should not start with (data)	The result of command "type file.txt" should not start with "abc"
System	CommandResultEndWith	The result of command (data) should end with (data)	The result of command "type file.txt" should end with "abc"
System	CommandResultDoesNotEndWith	The result of command (data) should not end with (data)	The result of command "type file.txt" should not end with "abc"
System	CommandResultContains	The result of command (data) should contain (data)	The result of command "type file.txt" should contain "abc"
System	CommandResultDoesNotContain	The result of command (data) should not contain (data)	The result of command "type file.txt" should not contain "abc"
System	CommandResultPatternMatch	The result of command (data) should match pattern (data)	The result of command "type file.txt" should match pattern "N2"
System	CommandResultPatternNotMatch	The result of command (data) should not match pattern (data)	The result of command "type file.txt" should not match pattern "N2"
System	CommandResultRegexMatch	The result of command (data) should match (data)	The result of command "type file.txt" should match "abc"
System	CommandResultRegexNotMatch	The result of command (data) should not match (data)	The result of command "type file.txt" should not match "abc"

Appendix B – Email Keywords

Actions

Category	Name	Description	Example
Email	SetEmailSMTPHost	Set email SMTP Host with value (data)	Clarity sets email SMTP Host with value clarity.com
Email	SetEmailSMTPPort	Set email SMTP Port with value (data)	Clarity sets email SMTP PORT with value 25
Email	SetEmailSMTPUsername	Set email SMTP Username with value (data)	Clarity sets email SMTP Username with value userEmail
Email	SetEmailSMTPPassword	Set email SMTP Password with value (data)	Clarity sets email SMTP Password with value UserPassword
Email	SetEmailSMTPEncryption	Set email SMTP encryption with value (data)	Clarity sets email SMTP encryption with value SSL
Email	SetEmailSubject	Set email subject with value (data)	Clarity sets email subject with value "Clarity"
Email	SetEmailContent	Set email content with value (data)	Clarity sets email content with value "Hello Clarity"
Email	SetEmailFrom	Set email from with value (data)	Clarity sets email from with value clarity@clarity.com
Email	SetEmailDisplayFrom	Set email displayfrom with value (data)	Clarity sets email displayfrom with value Clarity
Email	SetEmailReplyTo	Set email replyto with value (data)	Clarity sets email replyto with value clarity@clarity.com
Email	SendEmail	Send email to (data)	Clarity sends email to mailbox@server.com
Email	SelectTabName	selects tab name (data)	The user selects tab name cnn
Email	SetEmailPOP3Host	Set email POP3 Host with value (data)	Clarity sets email POP3 Host with value clarity.com
Email	SetEmailPOP3Port	Set email POP3 Port with value (data)	Clarity sets email POP3 Port with value 25
Email	SetEmailPOP3Username	Set email POP3 Username with value (data)	Clarity sets email POP3 Username with value userEmail
Email	SetEmailPOP3Password	Set email POP3 Password with value (data)	Clarity sets email POP3 Password with value UserPassword
Email	SetEmailPOP3Encryption	Set email POP3 encryption with value (data)	Clarity sets email POP3 encryption with value SSL
Email	DeletePOP3Email	Delete email using POP3	Clarity deletes email using POP3
Email	ResetEmailFilter	Reset email filter	the user resets the email filter
Email	SetEmailFilterSubject	Set subject of email filter with value (data)	The user sets subject of email filter with value "Clarity"
Email	SetEmailFilterContent	Set content of email filter with value (data)	The user sets content of email filter with value "Clarity"
Email	SetEmailFilterSentDate	Set sentdate of email filter with value (data)	The user sets sentdate of email filter with value "2022-04-11"
Email	SetEmailFilterFrom	Set from of email filter with value (data)	Set from of email filter with value (data)
Email	SetEmailFilterDisplayFrom	Set displayfrom of email filter with value (data)	Set displayfrom of email filter with value (data)
Email	SetEmailFilterReplyTo	Set replyto of email filter with value (data)	Set replyto of email filter with value (data)
Email	SetDataWithEmailCount	Sets data (data) with email count	the user sets data data1 with email count
Email	SetDataWithEmailSubject	Sets data (data) with email subject	the user sets data data1 with email subject
Email	SetDataWithEmailTo	Sets data (data) with email to	the user sets data data1 with email to
Email	SetDataWithEmailContent	Sets data (data) with email content	the user sets data data1 with email content
Email	SetDataWithEmailFrom	Sets data (data) with email from	the user sets data data1 with email from
Email	SetDataWithEmailDisplayFrom	Sets data (data) with email displayfrom	the user sets data data1 with email displayfrom
Email	SetDataWithEmailReplyTo	Sets data (data) with email replyto	the user sets data data1 with email replyto
Email	SetDataWithEmailSentDate	Sets data (data) with email sentdate	the user sets data data1 with email sentdate
Email	SetGlobalDataWithEmailCount	Sets global data (data) with email count	the user sets global data data1 with email count
Email	SetGlobalDataWithEmailSubject	Sets global data (data) with email subject	the user sets global data data1 with email subject
Email	SetGlobalDataWithEmailTo	Sets global data (data) with email to	the user sets global data data1 with email to
Email	SetGlobalDataWithEmailContent	Sets global data (data) with email content	the user sets global data data1 with email content
Email	SetGlobalDataWithEmailFrom	Sets global data (data) with email from	the user sets global data data1 with email from
Email	SetGlobalDataWithEmailDisplayFrom	Sets global data (data) with email displayfrom	the user sets global data data1 with email displayfrom
Email	SetGlobalDataWithEmailReplyTo	Sets global data (data) with email replyto	the user sets global data data1 with email replyto
Email	SetGlobalDataWithEmailSentDate	Sets global data (data) with email sentdate	the user sets global data data1 with email sentdate

Validations

Category	Name	Description	Example
Email	EmailCountEqualTo	The count of email should be equal to (data)	The count of email should be equal to 1
Email	EmailCountNotEqualTo	The count of email should be not equal to (data)	The count of email should be not equal to 1
Email	EmailCountGreaterThan	The count of email should be greater than (data)	The count of email should be greater than 1
Email	EmailCountGreaterThanOrEqual	The count of email should be greater or equal to (data)	The count of email should be greater or equal to 1
Email	EmailCountLowerThan	The count of email should be lower than (data)	The count of email should be lower than 1
Email	EmailCountLowerThanOrEqual	The count of email should be lower or equal to (data)	The count of email should be lower or equal to 1
Email	EmailContentEqualTo	The Email content should be equal to (data)	The Email content should be equal to "abc"
Email	EmailContentNotEqualTo	The Email content should be not equal to (data)	The Email content should be not equal to "abc"
Email	EmailContentStartWith	The Email content should start with (data)	The Email content should start with "abc"
Email	EmailContentNotStartWith	The Email content should not start with (data)	The Email content should not start with "abc"
Email	EmailContentEndWith	The Email content should end with (data)	The Email content should end with "abc"
Email	EmailContentNotEndWith	The Email content should not end with (data)	The Email content should not end with "abc"
Email	EmailContentContains	The Email content should contain (data)	The Email content should contain "abc"
Email	EmailContentNotContain	The Email content should not contain (data)	The Email content should not contain "abc"
Email	EmailContentPatternMatch	The Email content should match pattern (data)	The Email content should match pattern N5
Email	EmailContentPatternNotMatch	The Email content should not match pattern (data)	The Email content should not match pattern N5
Email	EmailContentRegExMatch	The Email content should match (data)	The Email content should match "abc"
Email	EmailContentRegExNotMatch	The Email content should not match (data)	The Email content should not match "abc"

Appendix C – API Keywords

Actions

Category	Name	Description	Example
API	SetAPIMethod	Set API method to (data)	The call sets API method to GET
API	SetAPIHeader	Set API Header (attribute) with value (data)	The call sets API Header SessionID with value ABC123
API	DeleteAPIHeader	Delete API Header (attribute)	The call deletes API Header SessionID
API	DeleteAllAPIHeaders	Delete All API Headers	The call deletes all API Headers
API	SetAPIHeaderFromResponse	Set API Header (attribute) from response	The call sets API Header SessionID from response
API	DeleteAPIHeaderFromResponse	Delete API Header (attribute) from response	The call deletes API Header SessionID from response
API	DeleteAllAPIHeadersFromResponse	Deletes All API Headers from response	The call deletes All API Headers from response
API	SetBasicAuthentication	Set API Basic Authentication with user: (data1) and password: (data2)	The call sets API basic authentication with user: username and password: password
API	ClearAuthentication	Clear API authentication	The call clears API authentication
API	SetAPIParameter	Set API Parameter (attribute) with value (data)	The call sets API Parameter ProductID with value 123
API	DeleteAPIParameter	Delete API Parameter (attribute)	The call deletes API Parameter ProductID
API	DeleteAllAPIParameters	Delete All API Parameters	The call deletes All API Parameters
API	SetAPIPayload	Set API Payload with value (data)	The call sets API Payload with value {ProductID: 123}
API	ClearAPIPayload	Clear API Payload	The call clears API Payload
API	CallAPI	Call API end point (data)	The system calls API end point http://server.com/api/products
Category	Name	Description	Example
Cookie-API	SetAPICookieFromBrowser	Set API Cookie (data) from browser	The call sets API Cookie session_id from browser
Cookie-API	SetAPICookieForDomainAndPath	Set API Cookie (cookie) with value (data1) for domain (data2) and path (data3)	The call sets API Cookie SessionID with value abc123 for domain google.com and path "/"
Cookie-API	DeleteAPICookieOfDomainAndPath	Delete API Cookie (cookie) of domain (data1) and path (data2)	The call deletes API Cookie SessionID of domain google.com and path "/"
Cookie-API	DeleteAllAPICookiesOfDomain	Delete All API Cookies of domain (data)	The call deletes All API Cookies of domain "google.com"
Cookie-API	DeleteAllAPICookiesOfPath	Delete All API Cookies of path (data)	The call deletes All API Cookies of path "/"
Cookie-API	DeleteAllAPICookies	Delete All API Cookies	The call deletes All API Cookies
Category	Name	Description	Example
Data-API	SetDataWithAPIReturnCode	Sets data (data) with API return code	The system sets data ReturnCode with API return code
Data-API	SetDataWithAPIReturnHeader	Sets data (data) with API return header attribute (attribute)	the system sets data SessionID with API return header attribute SessionID
Data-API	SetDataWithAPICookieValue	Sets data (data) with value of API cookie (cookie)	the system sets data SessionID with value of API cookie SessionID
Data-API	SetDataWithAPICookieDomain	Sets data (data) with domain of API cookie (cookie)	the system sets data SessionID with domain of API cookie SessionID
Data-API	SetDataWithAPICookiePath	Sets data (data) with path of API cookie (cookie)	the system sets data SessionID with path of API cookie SessionID
Data-API	SetDataWithAPICookieExpiryDate	Sets data (data) with expiry date of API cookie (cookie)	the system sets data SessionID with expiry date of API cookie SessionID
Data-API	SetDataWithAPIJSONReturnElement	Sets data (data) with API return JSON payload element (attribute)	The system sets data ProductID with API return JSON payload element Products[0].ID
Data-API	SetDataWithFullAPIReturnPayload	Sets data (data) with full API return payload	The system sets data ProductPayload with full API return payload
Data-API	SetGlobalDataWithAPIReturnCode	Sets global data (data) with API return code	The system sets global data ReturnCode with API return code
Data-API	SetGlobalDataWithAPIReturnHeader	Sets global data (data) with API return header attribute (attribute)	the system sets global data SessionID with API return header attribute SessionID
Data-API	SetGlobalDataWithAPICookieValue	Sets global data (data) with value of API cookie (cookie)	the system sets global data SessionID with value of API cookie SessionID
Data-API	SetGlobalDataWithAPICookieDomain	Sets global data (data) with domain of API cookie (cookie)	the system sets global data SessionID with domain of API cookie SessionID
Data-API	SetGlobalDataWithAPICookiePath	Sets global data (data) with path of API cookie (cookie)	the system sets global data SessionID with path of API cookie SessionID
Data-API	SetGlobalDataWithAPICookieExpiryDate	Sets global data (data) with expiry date of API cookie (cookie)	the system sets global data SessionID with expiry date of API cookie SessionID
Data-API	SetGlobalDataWithAPIJSONReturnElement	Sets global data (data) with API return JSON payload element (attribute)	The system sets global data ProductID with API return JSON payload element Products[0].ID
Data-API	SetGlobalDataWithFullAPIReturnPayload	Sets global data (data) with full API return payload	The system sets global data ProductPayload with full API return payload
Category	Name	Description	Example
Clipboard	SetClipboardWithAPIReturnCode	Sets clipboard with API return code	The system sets clipboard ReturnCode with API return code
Clipboard	SetClipboardWithAPIReturnHeader	Sets clipboard with API return header attribute (attribute)	the system sets clipboard SessionID with API return header attribute SessionID
Clipboard	SetClipboardWithAPICookieValue	Sets clipboard with value of API cookie (cookie)	Sets clipboard with value of API cookie accessToken
Clipboard	SetClipboardWithAPICookieDomain	Sets clipboard with domain of API cookie (cookie)	Sets clipboard with domain of API cookie accessToken
Clipboard	SetClipboardWithAPICookiePath	Sets clipboard with path of API cookie (cookie)	Sets clipboard with path of API cookie accessToken
Clipboard	SetClipboardWithAPICookieExpiryDate	Sets clipboard with expiry date of API cookie (cookie)	Sets clipboard with expiry date of API cookie accessToken
Clipboard	SetClipboardWithAPIJSONReturnElement	Sets clipboard with API return JSON payload element (attribute)	The system sets clipboard with API return JSON payload element Products[0].ID
Clipboard	SetClipboardWithFullAPIReturnPayload	Sets clipboard (clipboard) with full API return payload	The system sets clipboard ProductPayload with full API return payload

Validations

Category	Name	Description	Example
API	WebReturnCodeEqualTo	the return code of URL (data) should be equal to (data)	the return code of URL "http://www.google.com" should be equal to 200
API	WebReturnCodeNotEqualTo	the return code of URL (data) should be not equal to (data)	the return code of URL "http://www.google.com" should be not equal to 404
API	APIReturnCodeEqualTo	the API call return code should be equal to (data)	the API call return code should be equal to 200
API	APIReturnCodeNotEqualTo	the API call return code should be not equal to (data)	the API call return code should be not equal to 200
API	APIReturnHeaderElementExist	the API return header element (attribute) should exist	the API return header element SessionID should exist
API	APIReturnHeaderElementNotExist	the API return header element (attribute) should not exist	the API return header element SessionID should not exist
API	APIReturnHeaderElementEqualTo	the API response header element (attribute) should be equal to (data)	the API response header element "SessionID" should be equal to "session-1234"
API	APIReturnHeaderElementNotEqualTo	the API response header element (attribute) should be not equal to (data)	the API response header element "SessionID" should be not equal to "session-1234"
API	APIReturnHeaderElementContains	the API return header element (attribute) should contain (data)	the API return header element (attribute) should contain (data)
API	APIReturnHeaderElementNotContain	the API return header element (attribute) should not contain (data)	the API return header element (attribute) should not contain (data)
API	APIJSONReturnElementExist	the API response JSON payload element (attribute) should exist	the API response JSON payload element name should exist
API	APIJSONReturnElementNotExist	the API response JSON payload element (attribute) should not exist	the API response JSON payload element name should not exist
API	APIJSONReturnElementEqualType	the API return JSON payload element (attribute) should be equal to type (data)	the API return JSON payload element (element) should be equal to type (data)
API	APIJSONReturnElementNotEqualType	the API return JSON payload element (attribute) should be not equal to type (data)	the API return JSON payload element (element) should be not equal to type (data)
API	APIJSONReturnElementEqualTo	the API response JSON payload element (attribute) should be equal to (data)	the API response JSON payload element (element) should be equal to (data)
API	APIJSONReturnElementNotEqualTo	the API response JSON payload element (attribute) should be not equal to (data)	the API response JSON payload element (element) should be not equal to (data)
API	APIJSONReturnEqualTo	the API return JSON payload should be equal to (data)	the API return JSON payload should be equal to (data)
API	APIJSONReturnNotEqualTo	the API return JSON payload should be not equal to (data)	the API return JSON payload should be not equal to (data)
API	APIJSONReturnContains	the API return JSON payload should contain (data)	the API return JSON payload should contain (data)
API	APIJSONReturnNotContain	the API return JSON payload should not contain (data)	the API return JSON payload should not contain (data)

Appendix D – Visual Keywords

Actions

Category	Name	Description	Example
Data	SetDataWithObjectVisibility	Sets data (data) with visibility of object (object)	the user sets data data1 with visibility of object username
Data	SetDataWithObjectWidth	Sets data (data) with width of object (object)	the user sets data data1 with width of object username
Data	SetDataWithObjectHeight	Sets data (data) with height of object (object)	the user sets data data1 with height of object username
Data	SetDataWithObjectCoordX	Sets data (data) with coordinate X of object (object)	the user sets data data1 with coordinate X of object username
Data	SetDataWithObjectCoordY	Sets data (data) with coordinate Y of object (object)	the user sets data data1 with coordinate Y of object username
Data	SetDataWithPageImage	Sets data (data) with page image	Sets data PageImage with page image
Data	SetDataWithExtendedPageImage	Sets data (data) with extended page image	Sets data PageImage with extended page image
Data	SetDataWithObjectImage	Sets data (data) with image of object (object)	Sets data ButtonImage with image of object buttonOK
Data	SetGlobalDataWithObjectVisibility	Sets global data (data) with visibility of object (object)	the user sets global data data1 with visibility of object username
Data	SetGlobalDataWithObjectWidth	Sets global data (data) with width of object (object)	the user sets global data data1 with width of object username
Data	SetGlobalDataWithObjectHeight	Sets global data (data) with height of object (object)	the user sets global data data1 with height of object username
Data	SetGlobalDataWithObjectCoordX	Sets global data (data) with coordinate X of object (object)	the user sets global data data1 with coordinate X of object username
Data	SetGlobalDataWithObjectCoordY	Sets global data (data) with coordinate Y of object (object)	the user sets global data data1 with coordinate Y of object username
Data	SetGlobalDataWithPageImage	Sets global data (data) with page image	Sets global data PageImage with page image
Data	SetGlobalDataWithExtendedPageImage	Sets global data (data) with extended page image	Sets global data PageImage with extended page image
Data	SetGlobalDataWithObjectImage	Sets global data (data) with image of object (object)	Sets global data ButtonImage with image of object buttonOK
Category	Name	Description	Example
Clipboard	SetClipboardWithObjectVisibility	Sets clipboard with visibility of object (object)	the user sets clipboard with visibility of object username
Clipboard	SetClipboardWithObjectAttribute	Sets clipboard with attribute (attribute) of object (object)	The user sets clipboard with attribute name of object username
Clipboard	SetClipboardWithObjectAttributeURLReturnCode	Sets clipboard with return code of URL from attribute (data) of object (object)	The system sets clipboard with return code of URL from attribute href of object link
Clipboard	SetClipboardWithObjectStyleAttribute	Sets clipboard with style attribute (attribute) of object (object)	The user sets clipboard with style attribute color of object username
Clipboard	SetClipboardWithObjectWidth	Sets clipboard with width of object (object)	the user sets clipboard with width of object username
Clipboard	SetClipboardWithObjectHeight	Sets clipboard with height of object (object)	the user sets clipboard with height of object username
Clipboard	SetClipboardWithObjectCoordX	Sets clipboard with coordinate X of object (object)	the user sets clipboard with coordinate X of object username
Clipboard	SetClipboardWithObjectCoordY	Sets clipboard with coordinate Y of object (object)	the user sets clipboard with coordinate Y of object username

Validations

Category	Name	Description	Example
Browser	TabNamesVisible	tab name (data) should be visible	the tab name ibm should be visible
Browser	TabIsVisible	tab (data) should be visible	the tab MainTab should be visible
Browser	TabNamesNotVisible	tab name (data) should be not visible	the tab name ibm should be not visible
Browser	TabIsNotVisible	tab (data) should be not visible	the tab MainTab should be not visible
Browser	PageImageEqualTo	the page image should be equal to (data)	the page image should be equal to PageImage
Browser	PageImageNotEqualTo	the page image should be not equal to (data)	the page image should be not equal to PageImage
Browser	ExtendedPageImageEqualTo	the extended page image should be equal to (data)	the extended page image should be equal to PageImage
Browser	ExtendedPageImageNotEqualTo	the extended page image should be not equal to (data)	the extended page image should be not equal to PageImage
Category	Name	Description	Example
Object	IsVisible	object (object) should be visible	The object buttonOK should be visible
Object	IsNotVisible	object (object) should be not visible	The object buttonOK should be not visible
Object	IsEnabled	object (object) should be enabled	The object buttonOK should be enabled
Object	IsNotEnabled	object (object) should be not enabled	The object buttonOK should be not enabled
Object	AttributeEqualTo	Attribute (attribute) of object (object) should be equal to (data)	the attribute "value" of object "username" should be equal to "mike"
Object	AttributeNotEqualTo	Attribute (attribute) of object (object) should be not equal to (data)	the attribute "value" of object "username" should be not equal to "mike"
Object	AttributeStartWith	Attribute (attribute) of object (object) should start with (data)	the attribute "value" of object "username" should start with "mike"
Object	AttributeNotStartWith	Attribute (attribute) of object (object) should not start with (data)	the attribute "value" of object "username" should not start with "mike"
Object	AttributeEndWith	Attribute (attribute) of object (object) should end with (data)	the attribute "value" of object "username" should end with "mike"
Object	AttributeNotEndWith	Attribute (attribute) of object (object) should not end with (data)	the attribute "value" of object "username" should not end with "mike"
Object	AttributeContains	Attribute (attribute) of object (object) should contain (data)	the attribute "value" of object "username" should contain "mike"
Object	AttributeNotContain	Attribute (attribute) of object (object) should not contain (data)	the attribute "value" of object "username" should not contain "mike"
Object	AttributeRegExMatch	Attribute (attribute) of object (object) should match (data)	the attribute "value" of object "username" should match "m.*"
Object	AttributeRegExNotMatch	Attribute (attribute) of object (object) should not match (data)	the attribute "value" of object "username" should not match "m.*"
Object	AttributePatternMatch	Attribute (attribute) of object (object) should match pattern (data)	the attribute "value" of object "username" should match pattern "N2"
Object	AttributePatternNotMatch	Attribute (attribute) of object (object) should not match pattern (data)	the attribute "value" of object "username" should not match pattern "N2"
Object	StyleAttributeEqualTo	Style attribute (attribute) of object (object) should be equal to (data)	the style attribute "color" of object "username" should be equal to "mike"
Object	StyleAttributeNotEqualTo	Style attribute (attribute) of object (object) should be not equal to (data)	the style attribute "color" of object "username" should be not equal to "mike"
Object	StyleAttributeStartWith	Style attribute (attribute) of object (object) should start with (data)	the style attribute "color" of object "username" should start with "mike"
Object	StyleAttributeNotStartWith	Style attribute (attribute) of object (object) should not start with (data)	the style attribute "color" of object "username" should not start with "mike"
Object	StyleAttributeEndWith	Style attribute (attribute) of object (object) should end with (data)	the style attribute "color" of object "username" should end with "mike"
Object	StyleAttributeNotEndWith	Style attribute (attribute) of object (object) should not end with (data)	the style attribute "color" of object "username" should not end with "mike"
Object	StyleAttributeContains	Style attribute (attribute) of object (object) should contain (data)	the style attribute "color" of object "username" should contain "mike"
Object	StyleAttributeNotContain	Style attribute (attribute) of object (object) should not contain (data)	the style attribute "color" of object "username" should not contain "mike"
Object	StyleAttributePatternMatch	Style attribute (attribute) of object (object) should match pattern (data)	the style attribute "color" of object "username" should match pattern "N2"
Object	StyleAttributePatternNotMatch	Style attribute (attribute) of object (object) should not match pattern (data)	the style attribute "color" of object "username" should not match pattern "N2"
Object	StyleAttributeRegExMatch	Style attribute (attribute) of object (object) should match (data)	the style attribute "color" of object "username" should match "m.*"
Object	StyleAttributeRegExNotMatch	Style attribute (attribute) of object (object) should not match (data)	the style attribute "color" of object "username" should not match "m.*"
Object	WidthEqualTo	Width of objects (list of objects) should be equal to (data) px	the width of object [buttonOK, buttonCancel] should be equal to 45 px
Object	HeightEqualTo	Height of objects (list of objects) should be equal to (data) px	the height of object [buttonOK, buttonCancel] should be equal to 45 px
Object	SameWidth	objects (list of objects) should be of the same width	the objects [buttonOK, buttonCancel] should be of the same width
Object	SameHeight	objects (list of objects) should be of the same height	the objects [buttonOK, buttonCancel] should be of the same height
Object	SameSize	objects (list of objects) should be of the same size	the objects [buttonOK, buttonCancel] should be of the same size
Object	AreAligned	objects (list of objects) should be aligned [Left, Right, Top, Bottom, Center, HCenter, VCenter]	the objects [buttonOK, buttonCancel] should be aligned left
Object	AreAlignedMatrix	objects (list of objects) should be aligned in a matrix with rows of size (data)	objectsproducts should be aligned in a matrix with rows of size 4
Object	IsBelow	object (object1) should be below object (object2) by (data) px	the object buttonOK should be below object buttonCancel by 2 px
Object	IsAbove	object (object1) should be above object (object2) by (data) px	the object buttonOK should be above object buttonCancel by 2 px
Object	IsLeftOf	object (object1) should be left of object (object2) by (data) px	the object buttonOK should be left of object buttonCancel by 2 px
Object	IsRightOf	object (object1) should be right of object (object2) by (data) px	the object buttonOK should be right of object buttonCancel by 2 px
Object	ObjectImageEqualTo	the image of object (object) should be equal to (data)	the image of object buttonOK should be equal to ButtonImage
Object	ObjectImageNotEqualTo	the image of object (object) should be not equal to (data)	the image of object buttonOK should be not equal to ButtonImage
Category	Name	Description	Example
Data-Image	DataImageEqualTo	the image in data (data) should be equal to (data)	the image in data buttonOK should be equal to ButtonImage
Data-Image	DataImageNotEqualTo	the image in data (data) should be not equal to (data)	the image in data buttonOK should be not equal to ButtonImage

Appendix E – Flow Keywords

Logic

Category	Name	Description	Example
Logic	LogicIF	@if (data)	@if \${{Amout}} > 10)
Logic	LogicELSE	@else	@else
Logic	LogicENDIF	@endif	@endif

Loop

Category	Name	Description	Example
Loop	LoopRepeat	@repeat	@repeat
Loop	LoopUntil	@until (data)	@until \${{Amout}} > 10)
Loop	LoopUntilObjectExist	@untilobjectexist (object)	@untilobjectexist errorMsg
Loop	LoopUntilObjectNotExist	@untilobjectnotexist (object)	@untilobjectnotexist errorMsg
Loop	LoopWhile	@while (data)	@while \${{Amout}} > 10)
Loop	LoopEndWhile	@endwhile	@endwhile
Loop	LoopWhileObjectExist	@whileobjectexist (object)	@whileobjectexist errorMsg
Loop	LoopWhileObjectNotExist	@whileobjectnotexist (object)	@whileobjectnotexist errorMsg
Loop	LoopForEachObject	@foreachobject (object)	@foreachobject buttonDelete
Loop	LoopEndFor	@endfor	@endfor
Loop	LoopBreak	@break	@break
Loop	LoopContinue	@continue	@continue

Examples

- The user sets data data1 with value "10"
The user sets data data2 with value 20
@if \${{data1}} == 10)
The user sets the value {data1} in object TextInput1
@else
The user sets the value {data2} in object TextInput1
@endif
The value of object TextInput1 should be equal to {data1}
- The user sets data counter with value 0
@Repeat
The user sets data counter with value \${{counter}} + 1)
@Until \${{counter}} == 2)
the value of data counter should be equal to 2
- The user sets data counter1 with value 0
The user sets data total with value 0
@while \${{counter1}} < 5)
The user sets data total with value \${{total}} + 1)
The user sets data counter1 with value \${{counter1}} + 1)
@endwhile
the value of data total should be equal to 5
- The user sets data counter1 with value 0
The user clicks on object buttonClick
@whileobjectExist buttonClickMsg
The user clicks on object buttonClick
The user sets data counter1 with value \${{counter1}} + 1)
@endwhile
the value of data counter1 should be equal to 1

- *The user sets data counter1 with value 0
@whileobjectnotExist buttonClickMsg
The user clicks on object buttonClick
The user sets data counter1 with value $\$({counter1} + 1)$
@endwhile
the value of data counter1 should be equal to 1*
- *The user sets data counter1 with value 0
the user sets data counterObject with object count of InputTextObjects
@foreachobject InputTextObjects
The user sets data counter1 with value $\$({counter1} + 1)$
@endfor
the value of data counter1 should be equal to {counterObject}*

Appendix F – Keywords Data

The data used by all keywords can be originated from:

- *String Constant*
 - *Ex: "John Smith"*
- *Data content*
 - *Ex: {dataname}*
- *Data parameter content*
 - *Ex: <parameter name>*
- *Javascript execution*
 - *Ex: \$(javascript code)*

Appendix G – Options

Options are a way to alter the behaviour or the result of a keyword, either standard or business. Any number of options can be assigned to a keyword. This is done by adding the option at the end of a step using “{” and “}” as delimiters.

System Options

Option	Description	Example
AJAXTimeOut	Allows to define a specific timeout for the AJAX call associated with a test step. The timeout value is expressed in milliseconds.	The user clicks on object buttonOK {{AJAXTimeOut=2000}}
AlertAction	Allows a step action to be automatically followed by an interaction with a browser alert popup. The possible values are: OK and CANCEL	The user clicks on object buttonOK {{AlertAction=OK}}
AlertInput	Allow a step to insert a value in an alert popup	The user clicks on object buttonOK {{AlertAction=OK}} {{AlertInput=5}}
AlertOutputTo	Allow a step to store in a data the text shown on an alert popup	The user clicks on object buttonOK {{AlertAction=OK}} {{AlertOutputTo=VarAlert}}
AllowIncompleteMatrix	Allows a step to consider a non-complete object matrix	the objects products should be aligned in a matrix with rows of 4 {{AllowIncompleteMatrix}}
AllObjects	Allows a step to process all the objects return by the object mapping definition.	the value of object price should be equal to "£12.99" {{AllObjects}}
AllVisible	Allows a step to validate if an object is fully visible.	The object Tooltip should be visible {{AllVisible}}
Transform	Allows the framework to apply a predefined function to any retrieved value (Object, Data, etc...).	The user sets data filename1 with attribute href of object Image1 {{Transform=ExtractFileName}}
ColorAsRGBA	Allows the framework to convert color style attribute into RGBA expression: RGBA(red value, green value, blue value, Opacity value)	style attribute color of object StyleAttribute should be equal to RGBA(0, 0, 255, 1) {{ColorAsRGBA}}
CropArea	Allow to crop an area from an image capture. The area is defined by setting the coordinates (X, Y) of the top left and bottom right corners. If a coordinate is not set, the framework will automatically assign the following values: <ul style="list-style-type: none"> • Top Left X = 0 • Top Left Y = 0 • Bottom Right X = width of the image • Bottom Right Y = height of the image 	The user saves page image in data PageImage {{CropArea=10,10,20,20}}
DateFormat	Allows the framework to apply a format to a date value (Object, Data, etc...). The option can include a format as a parameter. If no parameter is used, the default format for the used locale is used.	the value of object price should be equal to "2022-01-02" {{DateFormat="yyyy-MM-dd"}}
Defect	Allows to associate one or more defects to a test step.	the value of object price should be equal to "12.99" {{Defect=Defect-1}} {{Defect=Defect-2}}
DeleteAllEmail	Allow to delete all emails with a unique step	Clarity deletes email using POP3 {{DeleteAllEmail}}
DisabledClassName	Allows to specify the name of a css class used to set an object disabled.	the object buttonOK should be not enabled {{DisabledClassName=disabled}}
EscapeDialog	Allows a step action to be automatically followed by an interaction with a browser popup with an escape action.	the user clicks on object AttachFile {{EscapeDialog}}
EmailIndex	Allow to define the index of the email to consider	the user sets data EmailSubject with email subject {{EmailIndex=1}}
ExcludedArea	Allow to exclude an area from an image capture. The area is defined by setting the coordinates (X, Y) of the top left and bottom right corners. If a coordinate is not set, the framework will automatically assign the following values: <ul style="list-style-type: none"> • Top Left X = 0 • Top Left Y = 0 • Bottom Right X = width of the image • Bottom Right Y = height of the image 	The user saves page image in data PageImage {{ExcludedArea=10,10,20,20}}
ExcludedObject	Allow to exclude objects from an image capture.	The user saves page image in data PageImage {{ExcludedObject=UnitPrice}}
Filter	Allows the framework to apply a regular expression to any value (Object, Data, etc...) and only consider the content of the first extracted group.	the value of object price should be equal to "12.99" {{Filter="£(.*)"}}
ForbidRedirect	Allows a step to not consider URL redirect when checking HTTP return code.	the return code of URL "http://www.google.com" should be equal to 200 {{ForbidRedirect}}
ForceKeystroke	Allow a step to simulate keystrokes when inserting values in an object.	the user sets the value john.smith@server.com in object Account.Email {{ForceKeystroke}}
ForceValueIntoField	Allow a step to force the value of an input field using Javascript.	the user sets the value john.smith@server.com in object Account.Email {{ForceValueIntoField}}
HomeDirectory	Allows to define the path of the home directory to use for a command execution	The system executes command dir {{HomeDirectory="C:\\"}}
IgnoreHeader	Allows to ignore the header information of data of type Table	The user saves data record in folder "target" with the file name "result.txt" {{IgnoreHeader}}
IterateOverParent	Allows a step to be iterated using a specific parent definition.	the objects [firstname, lastname] are aligned left {{IterateOverParent=TableRow}}
KeepMouseOver	Allows to force a step to maintain a previous mouse hover action.	The user clicks on object buttonOK {{KeepMouseOver}}
Locale	Allows the framework to apply localization rules as part of the data transformation process	The user set data TotalProductPrice with value 10.67 {{NumberFormat=0.00}} {{Locale=fr-fr}}
MultipleValue	Allows to configure a validation step so it can consider multiple values. The step execution is set as "Passed" when one of the value is validated.	the value of object price should be equal to ["£12.99", "Free"] {{MultipleValue}}
NumberFormat	Allows the framework to apply a format to a number value (Object, Data, etc...). The option can include a format as a parameter. If no parameter is used, the default format for the used locale is used.	the value of object OutputLocaleNumber should be equal to 123456.79 {{Locale=en-US}} {{NumberFormat}}
NoException	Allows a step to skip the use of test exceptions when an object is not located	the value of object price should be equal to "12.99" {{NoException}}
NotFoundColumn	Allows the framework to set a column to a table selection if no record is found. This option is mandatory if the option NotFoundValue is used	the user selects table PriceMapping where column ID is equal to Free {{NotFoundValue=0}} {{NotFoundColumn=Value}}

NotFoundValue	Allows the framework to set a value to a table selection if no record is found.	the user selects table PriceMapping where column ID is equal to Free <code>{{NotFoundValue=0}} {{NotFoundColumn=Value}}</code>
NoSessionSetup	Allows the framework to only run a step if no session setup steps have been executed.	the user selects table PriceMapping where column ID is equal to Free <code>{{NoSessionSetup}}</code>
ScreenShot	Allows to force not taking a screenshot after the execution of a step whatever the value of the step result.	the value of object price should be equal to "€12.99" <code>{{NoScreenshot}}</code>
NumberFormat	Allows the framework to apply a format to a numeric value (Object, Data, etc...).	the value of object price should be equal to "€12.99" <code>{{NumberFormat="€0.00"}}</code>
Object	Allows to force the usage of a specific object saved previously	the value of object price should be equal to "€12.99" <code>{{Object=SavedPrice}}</code>
ObjectAboveOfMapping	Allows to force the value of the attribute AboveOf in the object mapping definition.	the value of object price should be equal to "€12.99" <code>{{ObjectAboveOfMapping=ProductName}}</code>
ObjectBelowOfMapping	Allows to force the value of the attribute BelowOf in the object mapping definition.	the value of object price should be equal to "€12.99" <code>{{ObjectBelowOfMapping=ProductName}}</code>
ObjectChildMapping	Allows to force the value of the attribute Child in the object mapping definition.	the value of object price should be equal to "€12.99" <code>{{ObjectChildMapping=ProductName}}</code>
ObjectChildDepth	Allows to force the value of the attribute ChildDepth in the object mapping definition	the value of object price should be equal to "€12.99" <code>{{ObjectChildDepth=1}}</code>
ObjectContainer	Allows to force the value of the container object to use when applying the object mapping definition	the value of object price should be equal to "€12.99" <code>{{ObjectContainer=SavedContainer1}}</code>
ObjectContainerMapping	Allows to force the value of the attribute Container in the object mapping definition.	the value of object price should be equal to "€12.99" <code>{{ObjectContainerMapping=MainPanel}}</code>
ObjectContent	Allows to force the value of the attribute content in the object mapping definition	the value of object price should be equal to "€12.99" <code>{{ObjectContent="€"}}</code>
ObjectLeftOfMapping	Allows to force the value of the attribute LeftOf in the object mapping definition.	the value of object price should be equal to "€12.99" <code>{{ObjectLeftOfMapping=ProductName}}</code>
ObjectNearOfMapping	Allows to force the value of the attribute NearOf in the object mapping definition.	the value of object price should be equal to "€12.99" <code>{{ObjectNearOfMapping=ProductName}}</code>
ObjectRightOfMapping	Allows to force the value of the attribute RightOf in the object mapping definition.	the value of object price should be equal to "€12.99" <code>{{ObjectRightOfMapping=ProductName}}</code>
ObjectSelected	Allows to force the value of the attribute selected in the object mapping definition	The value of object price should be visible <code>{{ObjectSelected=True}}</code>
ObjectMultipleContent	Allows to force the value of the attribute multiplecontent in the object mapping definition	the value of object price should be equal to "€12.99" <code>{{ObjectMultipleContent="Content_US"}}</code>
ObjectDepth	Allows to force the value of the attribute depth in the object mapping definition	the value of object price should be equal to "€12.99" <code>{{ObjectDepth=1}}</code>
ObjectHidden	Allows to force the value of the attribute hidden in the object mapping definition	the value of object price should be equal to "€12.99" <code>{{ObjectHidden=true}}</code>
ObjectIndex	Allows to force the value of the attribute Index in the object mapping definition.	the value of object price should be equal to "€12.99" <code>{{ObjectIndex=2}}</code>
ObjectParent	Allows to force the value of the parent object to use when applying the object mapping definition	the value of object price should be equal to "€12.99" <code>{{ObjectParent=SavedParent1}}</code>
ObjectParentDepth	Allows to force the value of the attribute ParentDepth in the object mapping definition	the value of object price should be equal to "€12.99" <code>{{ObjectParentDepth=1}}</code>
ObjectParentIndex	Allows to force the value of the attribute ParentIndex in the object mapping definition This option can have second parameter with the name of an existing parent to which the option should be applied. If the second parameter is equal to \$MainParent, the option value will be used as the first parent to use for the object mapping.	the value of object price should be equal to "€12.99" <code>{{ObjectParentIndex=2}}</code> the value of object price should be equal to "€12.99" <code>{{ObjectParentIndex=2, subParent1}}</code> the value of object price should be equal to "€12.99" <code>{{ObjectParentIndex=2, \$MainParent}}</code>
ObjectParentMapping	Allows to force the value of the attribute Parent in the object mapping definition. This option can have second parameter with the name of an existing parent to which the option should be applied. If the second parameter is equal to \$MainParent, the option value will be used as the first parent to use for the object mapping.	the value of object price should be equal to "€12.99" <code>{{ObjectParentMapping=MainPanel}}</code> the value of object price should be equal to "€12.99" <code>{{ObjectParentMapping=MainPane, subParent1}}</code> the value of object price should be equal to "€12.99" <code>{{ObjectParentMapping=MainPanel, \$MainParent}}</code>
ObjectShadowParent	Allows to force the value of the attribute ShadowParent in the object mapping definition	the value of object price should be equal to "€12.99" <code>{{ObjectShadowParent=true}}</code>
ObjectSiblingDepth	Allows to force the value of the attribute SiblingDepth in the object mapping definition	the value of object price should be equal to "€12.99" <code>{{ObjectSiblingDepth=1}}</code>
ObjectType	Allows to force the value of the attribute type in the object mapping definition	the value of object price should be equal to "€12.99" <code>{{ObjectType=xpath}}</code>
ObjectValue	Allows to force the value of the attribute value in the object mapping definition	the value of object price should be equal to "€12.99" <code>{{ObjectValue=input#amount}}</code>
Offset	Allows to specify the offset position (X, Y) inside an object where hover and click actions are performed	The user clicks on object Table <code>{{Offset=600, 0}}</code>
Optional	Allows to consider a step as an optional step. The result of the step will not be considered to calculate the result of the test. The optional timeout value is expressed in milliseconds. If no value is specified, the default timeout value is used.	the value of object price should be equal to "€12.99" <code>{{Optional}}</code> the value of object price should be equal to "€12.99" <code>{{Optional=1000}}</code>
ParseNumber	Allow to parse a string as a number. The option can include a format as a parameter. If no parameter is used, the default format for the used locale is used.	the value of object OutputLocaleNumber should be equal to 123456.79 <code>{{Locale=en-US}} {{ParseNumber}}</code>
ParseDate	Allow to parse a string as a date. The option can include a format as a parameter. If no parameter is used, the default format for the used locale is used.	the value of object OutputLocaleNumber should be equal to 123456.79 <code>{{Locale=en-US}} {{ParseDate=yyyy-MM-dd}}</code>
PrettyFormat	Allows to force a pretty format conversion for the data	the API response JSON payload should be equal to <code>{APIPost_Response} {{PrettyFormat}}</code>
ProcessIDData	Allows to define the name of the data used to store the process ID of a command execution	The system executes command <code>npm start</code> <code>{{ProcessIDData=ProcID}}</code>
ReturnCodeData	Allows to define the name of the data used to store the return code of a command execution	The system executes command <code>dir</code> <code>{{ReturnCodeData=RC}}</code>
RowData	Allows to define the name of the data used when selecting a table data row	The user selects data Country with ID UK <code>{{RowData=UKRow}}</code>
SaveMode	Allows to define the file saving mode. The available values are: <ul style="list-style-type: none"> Overwrite (Default) Append 	The user saves data record in folder "target" with the file name "result.txt" <code>{{SaveMode=Append}}</code>

SaveFormat	Allows to define the file saving format. The available values are: <ul style="list-style-type: none"> • TXT • CSV • TSV • Excel 	The user saves data record in folder "target" with the file name "result.txt" <code>{{SaveFormat=Excel}}</code>
SaveObject	Allows a step to save an object information for later reuse.	The user hovers over object TargetObject <code>{{SaveObject=HoverObject}}</code>
SaveObjectParent	Allows a step to save object parent information for later reuse.	the objects [buttonOK, buttonCancel] are aligned left <code>{{SaveObjectParent=SavedParent1}}</code>
ScreenShot	Allows to force taking a screenshot after the execution of a step whatever the value of the step result.	the value of object price should be equal to "€12.99" <code>{{ScreenShot}}</code>
SkipCompletion	Allows a step to not wait for the completion of the execution of a command	The system executes command SetUpload <code>"\${FilePath.Clarity_TXT}" {{SkipCompletion}}</code>
SkipWaitForHTML	Allows to force a step to skip the wait for the full HTML page to be loaded	the object buttonOK should be not enabled <code>{{SkipWaitForHTML}}</code>
SlowInput	Allows to define a low input speed when inserting data. The framework will enter data one character at a time	the user sets the value "abc" in object username <code>{{SlowInput}}</code>
StringFormat	Allows the framework to apply a format to a string value (Object, Data, etc...). The format is defined by using # to match each character.	the value of object phoneNumber should be equal to "555 123 456 789" <code>{{StringFormat="### ### ### ###"}}</code>
TimeOut	Allows to define a specific timeout for a test step. The timeout value is expressed in milliseconds.	the value of object price should be equal to "12.99" <code>{{Timeout=1000}}</code>
NoTimeOut	Allows to run a step with no timeout.	the value of object price should be equal to "12.99" <code>{{NoTimeout}}</code>
VisualTolerance	Allows to force the value of the visual tolerance to use for a specific step.	the height of object buttonOK should be equal to 45 px <code>{{VisualTolerance=5}}</code>
WaitForAnimation	Allows a step to be executed after waiting for an object animation to complete	the object buttonOK should be not enabled <code>{{WaitForAnimation}}</code>
WaitForAJAX	Allows to force a step to wait for an AJAX call	the object buttonOK should be not enabled <code>{{WaitForAJAX}}</code>
WaitForObject	Allows to force a step to wait for an object to be visible. When this option is used in combination with the AllObjects option, a second parameter can be used to define how many iterations would skip the option.	The user clicks on object Delete <code>{{WaitForObject=ConfirmationMsg}}</code> The user clicks on object DeleteProduct <code>{{AllObjects}}</code> <code>{{WaitForObject=TotalAmount, 1}}</code>
Warning	Transforms the failure results of a step into a warning results without interrupting the test execution	the value of object price should be equal to "12.99" <code>{{Warning}}</code>

Custom Options

The Custom Options allow to pass information directly to a step for its execution

Option	Description	Example
<code>`\${Data name}`</code>	Allows to create or update a data to be used during the step execution	the value of object price should be equal to "12.99" <code>{{\${product_name="Hammer"}}</code>

Loop Options

The Loop Options allow to pass information directly to a loop step for its execution

Option	Description	Example
LoopIterationData	Allows to define the name of the data used to store the iteration of the loop	<code>@Repeat {{LoopIterationData=Loop1}}</code>
TimeOut	Allows to define a specific timeout for a test step. The timeout value is expressed in milliseconds.	<code>@UntilObjectExist ErrorMessage {{Timeout=1000}}</code>
NoTimeOut	Allows to run a step with no timeout.	<code>@UntilObjectExist ErrorMessage {{NoTimeout}}</code>

Option: Transform

This option allows the framework to apply a predefined function to any retrieved value (Object, Data, etc...).

The available functions are:

Function	Parameters	Description
ExtractFileFullName		Extract the file name (including the extension) from a string representing the full path file name. Example: The user sets data filename1 with attribute href of object Image1 {{Transform=ExtractFileFullName}}
ExtractFileName		Extract the file name from a string representing the full path file name. Example: The user sets data filename1 with attribute href of object Image1 {{Transform=ExtractFileName}}
ExtractFileExtension		Extract the file extension from a string representing the full path file name. Example: The user sets data filename1 with attribute href of object Image1 {{Transform=ExtractFileExtension}}
ExtractFilePath		Extract the file path from a string representing the full path file name. Example: The user sets data filename1 with attribute href of object Image1 {{Transform=ExtractFilePath}}
SubString	startIndex (> 0) endIndex (> 0) (Optional)	Extract a string from another string starting at position: startIndex and ending at position: endIndex. If endIndex does not exist, the return string contains all the characters from position startIndex. Example: The value of object CountrySelector.Banner.SelectedCountry should be equal to <Country> {{Transform=SubString,1,3}}
Replace	Original string Replacement string	Replace in a string, a targeted string with a replacement string. The targeted string is defined as a regular expression. Example: The value of object Product.Price should be equal to <ProductPrice> {{Transform=Replace, "_", "."}}
URLEncode		Encode string to be compatible as a URL. Example: {{Transform=URLEncode}}
URLDecode		Decode URL string. Example: {{Transform=URLDecode}}

Appendix H – Keys Characters

Keys characters are way for Clarity to interact with the application using special keystrokes. Those characters are only available with the keyword: types.

In order to send a combination of keystrokes at the same time, a list of keys and characters can be put in a comma separated list delimited by “[” and “]”.

Examples:

Select All (Ctrl + A): The user types the value [Keys.CONTROL, “A”] in object editfield

List of Keys Characters

- Keys.ADD
- Keys.ALT
- Keys.ARROW_DOWN
- Keys.ARROW_LEFT
- Keys.ARROW_RIGHT
- Keys.ARROW_UP
- Keys.BACK_SPACE
- Keys.CANCEL
- Keys.CLEAR
- Keys.COMMAND
- Keys.CONTROL
- Keys.DECIMAL
- Keys.DELETE
- Keys.DIVIDE
- Keys.DOWN
- Keys.END
- Keys.ENTER
- Keys.EQUALS
- Keys.ESCAPE
- Keys.F1
- Keys.F10
- Keys.F11
- Keys.F12
- Keys.F2
- Keys.F3
- Keys.F4
- Keys.F5
- Keys.F6
- Keys.F7
- Keys.F8
- Keys.F9
- Keys.HELP
- Keys.HOME
- Keys.INSERT
- Keys.LEFT
- Keys.LEFT_ALT
- Keys.LEFT_CONTROL
- Keys.LEFT_SHIFT
- Keys.META
- Keys.MULTIPLY
- Keys.NULL
- Keys.NUMPADO
- Keys.NUMPAD1
- Keys.NUMPAD2
- Keys.NUMPAD3

- *Keys.NUMPAD4*
- *Keys.NUMPAD5*
- *Keys.NUMPAD6*
- *Keys.NUMPAD7*
- *Keys.NUMPAD8*
- *Keys.NUMPAD9*
- *Keys.PAGE_DOWN*
- *Keys.PAGE_UP*
- *Keys.PAUSE*
- *Keys.RETURN*
- *Keys.RIGHT*
- *Keys.SEMICOLON*
- *Keys.SEPARATOR*
- *Keys.SHIFT*
- *Keys.SPACE*
- *Keys.SUBTRACT*
- *Keys.TAB*
- *Keys.UP*

Appendix I – Safari Configuration

In order to use the automation framework with a Safari browser, you need to use an Apple computer and allow the automation of the Safari browser by running the following command:

- *safaridriver --enable*

Appendix J – Local Mobile Configuration

In order to use the automation framework with a real device connected to your laptop, here are the steps to follow:

Android Devices

- *Set the device with USB Debugging mode activated*
- *Install ADB (Android Debug Bridge)*
- *Connect the device using USB*
- *Identify the device ID by running the command: adb devices*
- *Install Appium Desktop*
- *Start Appium Server*
- *Create a framework configuration file with a LocalDevice type of capability and setting the OS attribute to the value "Android".*

iOS Devices (Requires an Apple computer)

- *Install Appium Desktop*
- *Configure "Appium-WebDriverAgent" so it can be downloaded on the device by Appium. This step might require the compilation of the project WebDriverAgent.xcodeproj using the XCode application.*
- *Start Appium Server*
- *Create a framework configuration file with a LocalDevice type of capability and setting the OS attribute to the value "iOS".*

Appendix K – Path Management

In the configuration file, folder and file paths can be defined in a relative or an absolute manner.

Path Type	Definition															
Absolute	Absolute paths should start with the prefix: \$\$															
Relative to the Framework installation folder	Relative paths to the framework installation folder should start with the prefix: %%															
Relative to the attribute ReferencePath folder	Relative paths are relative to a reference folder. The reference folder is defined accordingly to the following decision table <table border="1" data-bbox="483 541 1344 741"> <thead> <tr> <th>Config.ReferencePath defined</th> <th>GitRepository.ReferencePath defined And the -gc run option active</th> <th>Reference Folder</th> </tr> </thead> <tbody> <tr> <td>Yes</td> <td>Yes</td> <td>GitRepository.ReferencePath</td> </tr> <tr> <td>Yes</td> <td>No</td> <td>Config.ReferencePath</td> </tr> <tr> <td>No</td> <td>Yes</td> <td>GitRepository.ReferencePath</td> </tr> <tr> <td>No</td> <td>No</td> <td>Clarity installation folder</td> </tr> </tbody> </table>	Config.ReferencePath defined	GitRepository.ReferencePath defined And the -gc run option active	Reference Folder	Yes	Yes	GitRepository.ReferencePath	Yes	No	Config.ReferencePath	No	Yes	GitRepository.ReferencePath	No	No	Clarity installation folder
Config.ReferencePath defined	GitRepository.ReferencePath defined And the -gc run option active	Reference Folder														
Yes	Yes	GitRepository.ReferencePath														
Yes	No	Config.ReferencePath														
No	Yes	GitRepository.ReferencePath														
No	No	Clarity installation folder														

Appendix L – Custom Keyword Development

The implementation of Custom keywords is a way to extend the capabilities of the framework by adding bespoke keyword. The logic associated with the custom keyword needs to be implemented in a Java Jar file. There is no constraint associated with the name of the Jar file, the Class name and the Method name containing the keyword logic, however those names should be referenced in the Custom Keyword definition in order for the framework to use them.

The only constraints for the implementation of the custom keyword are the signature of the method and the type of the return result.

Method Signature

The signature of the custom keyword method should contain all the following fields in the exact sequence.

Field Type	Description
<code>org.apache.log4j.Logger</code>	Reference to the framework logger component
<code>org.openqa.selenium.WebDriver</code>	Reference to the selenium webdriver. This component is only initialized if the custom keyword is associated with the platform WEB.
<code>ArrayList<String></code>	Reference to the list of parameters extracted from the custom keyword RegEx.
<code>TreeMap<String, ArrayList<String>></code>	Reference to the list of step options defined for the execution of the custom keyword
<code>ArrayList<ArrayList<org.openqa.selenium.WebElement>></code>	Reference to the list of objects defined with the custom keyword. The first dimension of the array correspond to the ObjectIndex dimension of the custom keyword. This list is only initialized if the custom keyword is associated with the platform WEB.

Method Return Type & Value

The custom keyword method should always return a comma separated String. The value before the first comma is associated to the return code of the step. The rest of the string is associated with an optional message to be displayed in the execution report.

The possible values for the return code are:

Return Code	Description
1	PASSED
2	FAILED
3	WARNING
4	ACCEPTABLE
5	PENDING

If the method returns an Exception, the framework will try to re-execute the step until reaching the timeout time limit.

Example

Here is an example of a custom keyword method inserting a string into a web input field.

```
import java.util.ArrayList;
import java.util.TreeMap;

import org.apache.log4j.Logger;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;

public class CustomKeyword_Class {
    public String customKeyword_Method(
        Logger testRunLogger,
        WebDriver webDriver,
        ArrayList<String> keywordParam,
        TreeMap<String, ArrayList<String>> tmOptions,
        ArrayList<ArrayList<WebElement>> keywordWebElements) throws Exception {

        String res = "0";

        WebElement object = keywordWebElements.get(0).get(0);

        testRunLogger.error("CustomKeyword-Insert -> ObjectName: "
            + keywordParam.get(1) + " - Value: " + keywordParam.get(0));

        try {
            object.sendKeys(keywordParam.get(0));
            res = "1";
        } catch (Throwable exc) {
            res = "2," + exc.getMessage();
        }

        return res;
    }
}
```

In this context, the custom keyword would be defined as followed:

```
CustomKeywords:
- RegEx: "CustomKeyword-Insert (.*) in (.*)"
Platform: WEB
ObjectIndex: 2
LibraryPath: "ClarityCustomKeyword-1.0.jar"
ClassName: com.CustomKeyword_Class
MethodName: customKeyword_Method
```

And its usage in a test scenario would be using a step such as:

CustomKeyword-Insert "abc" in "EditField-Object"

Appendix M – Data Transformation

During the execution of tests, transforming data is common and important to be flexible. For this purpose, Clarity includes a set of step options:

Option Name	Parameter	Example	Description
Filter	RegEx	Filter= order#\s(.*)\s	Allow to extract a subset of a string by using a RegEx. The extracted value is contained in the first regex group.
Locale	Locale value	En-GB	Allow to use localization rules
ParseNumber			Allow to parse a string into a number
ParseDate	Date format	yyyy-MM-dd	Allow to parse a string into a Date
StringFormat	String format	###.###.###.###	Allow to format a string using a pattern. Each # character is matched to the string character
NumberFormat	Number format	0.00	Allow to format a number using a format. If no format is provided the default locale format is used.
DateFormat	Date format	HH:mm:ss	Allow to format a date using a format. If no format is provided the default locale format is used.
Transform	ExtractFileFullName ExtractFileName ExtractFilePath SubString Replace URLEncode URLDecode		Apply to execute predefined functions
AddDays	Number	1	Allow to add a number of days to a date
AddMonths	Number	1	Allow to add a number of months to a date
AddYears	Number	1	Allow to add a number of years to a date
AddHours	Number	1	Allow to add a number of hours to a date
AddMinutes	Number	1	Allow to add a number of minutes to a date
AddSeconds	Number	1	Allow to add a number of seconds to a date

Examples: String Transformation

- *The user sets data varTransformation with value "Order# 1234 confirmed" {{Filter=#(.*)\s}}*
- *the value of data varTransformation should be equal to 1234*
- *The user sets data varTransformation with value 10,67 {{Locale=fr-FR}} {{StringFormat=## #}}*
- *the value of data varTransformation should be equal to 10 , 67*
- *The user sets data varTransformation with value 10,67 € {{Locale=fr-FR}} {{ParseNumber}} {{StringFormat=## #}}*
- *the value of data varTransformation should be equal to 10 . 67*

Examples: Number Transformation

- *The user sets data varTransformation with value 10.67 {{Locale=fr-FR}}*
- *the value of data varTransformation should be equal to 10.67*
- *The user sets data varTransformation with value 10,67 € {{Locale=fr-FR}} {{ParseNumber}}*
- *the value of data varTransformation should be equal to 10.67*
- *The user sets data varTransformation with value 10,67 {{Locale=fr-FR}} {{ParseNumber=0.00}}*
- *the value of data varTransformation should be equal to 10.67*
- *The user sets data varTransformation with value 1 000,67 € {{Locale=fr-FR}} {{ParseNumber}}*
- *the value of data varTransformation should be equal to 1000.67*
- *The user sets data varTransformation with value 1 000,67 € {{Locale=fr-FR}} {{ParseNumber}} {{NumberFormat}}*
- *the value of data varTransformation should be equal to 1 000,67 €*
- *The user sets data varTransformation with value 1 000,67 € {{Locale=fr-FR}} {{ParseNumber}} {{NumberFormat=0.000}}*
- *the value of data varTransformation should be equal to 1000,670*
- *The user sets data varTransformation with value 1000.67 {{Locale=fr-FR}} {{NumberFormat}}*
- *the value of data varTransformation should be equal to 1 000,67 €*
- *The user sets data varTransformation with value 1000.679 {{Locale=fr-FR}} {{NumberFormat}}*
- *the value of data varTransformation should be equal to 1 000,68 €*
- *The user sets data varTransformation with value 1000.679 {{NumberFormat}}*
- *the value of data varTransformation should be equal to 1,000.679*
- *The user sets data varTransformation with value 1000.679 {{Locale=fr-FR}} {{NumberFormat=0}}*
- *the value of data varTransformation should be equal to 1001*

- *The user sets data varTransformation with value 1000.679 {{Locale=fr-FR}} {{NumberFormat=0}}*
the value of data varTransformation should be equal to 1001
- *The user sets data varTransformation with value 1000.679 {{Locale=fr-FR}} {{NumberFormat=#,###.00}}*
the value of data varTransformation should be equal to 1 000,68

Examples: Date Transformation

- *The user sets data varTransformation with value 2022-01-18 19:55:18.802 {{Locale=fr-FR}} {{DateFormat}}*
the value of data varTransformation should be equal to 18 janv. 2022 19:55:18
- *The user sets data varTransformation with value 2022-01-18 19:55:18.802 {{Locale=en-GB}} {{DateFormat}}*
the value of data varTransformation should be equal to 18-Jan-2022 19:55:18
- *The user sets data varTransformation with value 2022-01-18 19:55:18.802 {{Locale=en-US}} {{DateFormat}}*
the value of data varTransformation should be equal to Jan 18, 2022 7:55:18 PM
- *The user sets data varTransformation with value 18-Jan-2022 19:55:18 {{Locale=en-GB}} {{ParseDate}} {{DateFormat=EEEE, dd MMMM, yyyy}}*
the value of data varTransformation should be equal to Tuesday, 18 January, 2022
- *The user sets data varTransformation with value 18 janv. 2022 19:55:18 {{Locale=fr-FR}} {{ParseDate}} {{DateFormat=EEEE, dd MMMM, yyyy}}*
the value of data varTransformation should be equal to mardi, 18 janvier, 2022
- *The user sets data varTransformation with value 18 janv. 2022 19:55:18 {{Locale=fr-FR}} {{ParseDate}}*
the value of data varTransformation should be equal to 2022-01-18 19:55:18.000
- *The user sets data varTransformation with value 2022-01-18 19:55:18 {{ParseDate=yyy-MM-dd}}*
the value of data varTransformation should be equal to 2022-01-18 00:00:00.000
- *The user sets data varTransformation with value 2022-01-18 19:55:18 {{Locale=fr-FR}} {{ParseDate=yyy-MM-dd}}*
the value of data varTransformation should be equal to 2022-01-18 00:00:00.000

Examples: Transformation Functions

- *The user sets data varTransformation with value "C:/MyDir/MyFolder/TestFile.txt" {{Transform=ExtractFileFullName}}*
the value of data varTransformation should be equal to TestFile.txt
- *The user sets data varTransformation with value "C:/MyDir/MyFolder/TestFile.txt" {{Transform=ExtractFileName}}*
the value of data varTransformation should be equal to TestFile
- *The user sets data varTransformation with value "C:/MyDir/MyFolder/TestFile.txt" {{Transform=ExtractFileExtension}}*
the value of data varTransformation should be equal to txt
- *The user sets data varTransformation with value "0123456789" {{Transform=SubString, 4, 6}}*
the value of data varTransformation should be equal to 345
- *The user sets data varTransformation with value "0123456789" {{Transform=SubString, 7, 7}}*
the value of data varTransformation should be equal to 6789
- *The user sets data varTransformation with value "0123456789" {{Transform=SubString, 7, 7}}*
the value of data varTransformation should be equal to 6789
- *The user sets data varTransformation with value "0123456789" {{Transform=Replace, 234, XXX}}*
the value of data varTransformation should be equal to 01XXX56789
- *The user sets data varTransformation with value "01234567890123456789" {{Transform=Replace, 234, XXX}}*
the value of data varTransformation should be equal to 01XXX5678901XXX56789
- *The user sets data varTransformation with value "http://testingwithclarity.com/Page Name.html" {{Transform=URLEncode}}*
the value of data varTransformation should be equal to http%3A%2F%2Ftestingwithclarity.com%2FPage+Name.html
- *The user sets data varTransformation with value http%3A%2F%2Ftestingwithclarity.com%2FPage+Name.html {{Transform=URLDecode}}*
the value of data varTransformation should be equal to http://testingwithclarity.com/Page Name.html