

SG-TAS-2.1

Table of contents

Introduction	6
Welcome	6
What's New	7
TAS-Prerequisite	8
Get Started	8
Installer	9
License	10
Property files	12
Reporting app WAR	22
Default Property	23
How To	27
Configure Eclipse	27
Web Recorder	32
How To Encrypt Password	44
Native App Recorder	45
Stop_3DX_TAS Browser Instance	62
Auto Browsers Driver Download	63
3DX Dataloader	63
Write Test Scripts	73
How to write Test Script/TestCase	73
How to use xsd suggestions from eclipse	75
Execute	75
Parallel Execution	76
Gracefully Shutdown	83
CSV Rules	86
Reports	86
Database and WebUI	86
CSV	90
TestNG	93
JIRA	96
JIRA XRAY Integration	96
JIRA Integration	106
JIRA Folder Structure	110
JIRA Execution Result	129
TAG</> Library	146
Web	146
Action	147
Assert	149
AssertAlert	154
AssertDeletion	156
AssertFileExists	156
AssertIndentedTableRow	157
AssertListValues	159
AssertOutlookMailHeader	162
AssertPageLoadTime	163
AssertDate	163
AssertXML	165

ApplySearch	168
CallWebService	169
CheckElement	170
CheckIndentedTableRow	174
ClickBackButton	175
ClickCategoryCommand	175
ClickDashboardCheckbox	176
ClickElement	177
ClickForwardButton	179
ClickGlobalActionsMenu	180
ClickGlobalToolsMenu	181
ClickHomeMenu	182
ClickMyDeskMenu	183
ClickOverrideLink	184
ClickPortalCommand	184
ClickRefreshButton	185
ClickWindowElement	186
CloseAllOtherWindow	187
CloseCurrentWindow:	188
CreateDashboard	188
CloseCurrentWindow	189
DeleteAllCookies	190
DeleteCurrentTabInDashboard	192
DownloadFile	193
DownloadFileUsingCommand	194
DownloadFileUsingIcon	195
DragAndDrop	197
DeleteDashboard	201
Deactive6wSearch	201
EditIndentedTableRow	202
Execute	207
ELSEIF	209
ELSE	211
ENDIF	212
FindElement	212
FilterSearchForm	215
FilterSearchPerson	217
GlobalSearch	218
GetValueFromCookies	220
HandleAlert	221
HighlightElement	222
IF	223
Include	225
InputText	225
InputWindowElement	228
Lifecycle	230
Login	230
Logout	232
MaximiseWindow	233
MaximizeAndMinimize	233

MqlAssert:	235
OpenActionBarMenu	237
OpenChooser	238
OpenCompassApp	239
OpenNewTabInDashboard	240
OpenNewToolbarInDashboard	241
OpenSearchResult	242
OpenURL	243
Option	244
OptionalStart and OptionalClose	246
OutlookLogin	247
OutlookLogout	248
Post & EndPost	248
PinWidgetToDashboard	250
Print	251
ResetSearch	252
ReadUserInput (Deprecated)	253
ReadXML	253
RegisterIndentedTableObjects	254
RegisterObject	256
RunMQL	259
Step & EndStep	260
ScrollToElement	261
SearchOutlookMail	264
SelectAndSubmit6WTagsSearchSelectAndSubmit6WTagsSearch	265
SelectColor	267
SelectDate	267
SelectContextMenuOption	271
SelectElement	272
SelectIndentedTableRow	273
SelectorOpenNewIndentedTableRow	280
SelectSecurityContext	281
SelectWindowRegion	282
SendKeys	283
Setstartwindow	284
ShowNotification	285
StartPerformanceLogs & StopPerformanceLogs	285
SwitchToContentFrame	286
SwitchToDefaultContent	286
SwitchToDetailsDisplayFrame	287
SwitchToFrame	287
SwitchToParentFrame	289
SwitchToParentWindow	289
SwitchToPortalDisplayFrame	290
SwitchToSlideInWindow	290
SwitchToStartWindow	291
SwitchToWindow	291
SetContent	293
SelectAndSubmit	294
UploadFileForDashboard	295

UploadFileUsingDragAndDrop	296
UploadFileUsingLocator	297
ValidateBegin	298
ValidateBehaviour	299
ValidateEnd	300
ValidateTableExport	301
ValidateTableHeader	303
Wait	303
RefreshCurrentPage	304
Native	305
3DX Tags	305
General Tags	310
ARAS	324
CheckBox	324
ClickItemViewCommandBar	325
ClickRelationshipTab	326
CloseNavigationPanel	326
LogIn	327
LogOut	328
OpenNavigationPanel	328
ShowContents	329
SwitchToFrameByTitle	330
UserMenu	331
OpenMainWindowApp	332
Jenkins Installation for SG- TAS-2.1	332
1. Introduction	332
2. Software Requirement	333
3. Steps	334
4. Components to deploy/configure	338

Introduction

Created with the Personal Edition of HelpNDoc: [Transform Your Help Documentation Process with a Help Authoring Tool](#)

Welcome to the world of test automation with STEEPGRAPH's promise of excellence. SG-TAS is a 3DEXPERIENCE focused set of tools which will help to minimize the effort and time with automated testing capabilities

Need For Automation Testing:

For all those customers who have been using ENOVIA/3DEXPERIENCE/ARAS platform, there is always nervousness before deploying a new set of features, bug fixes or DS Fix packs, whether these new changes will break any of the existing functionality. This means manual testing efforts are needed, but still, it is nearly impossible to test all areas of customization.

- Ensure there are no regressions post deploying new enhancements/bugs/fix packs
- Achieve comprehensive test coverage
- Optimize cost and efforts required to achieve target coverage
- Ability to measure and quantify software quality
- Reduce testing timeline

Scope of Software Testing Automation in PLM:

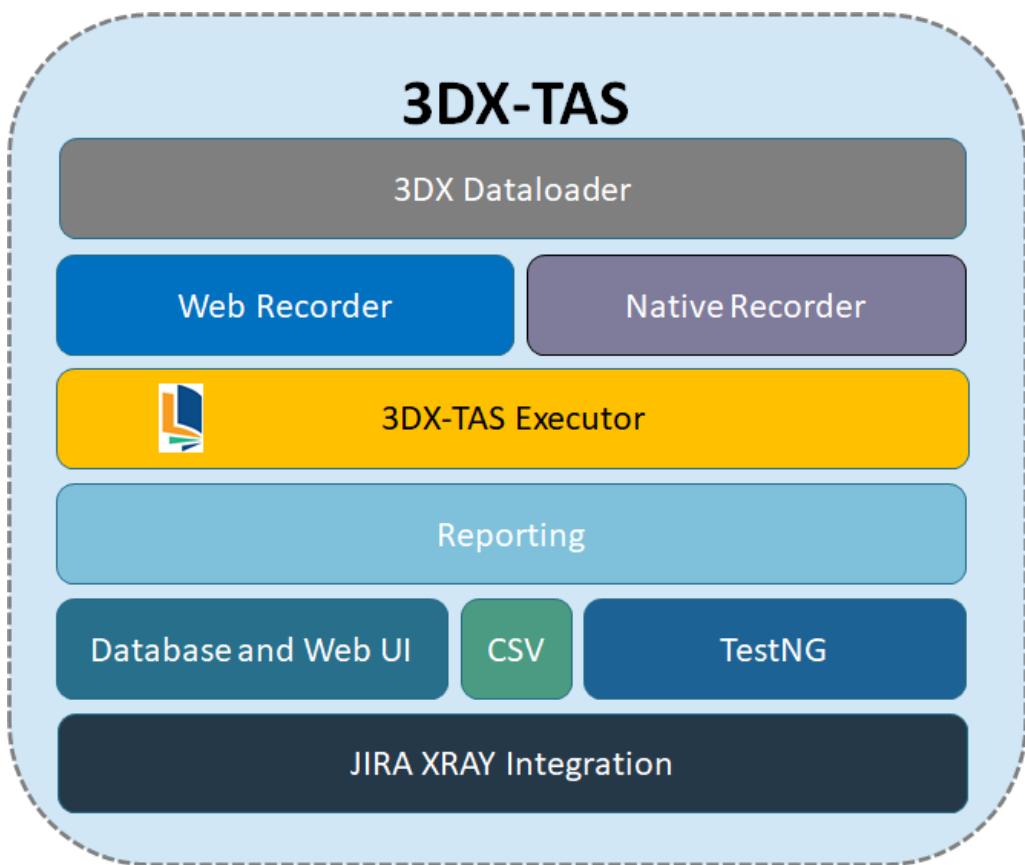
- Functional Testing
- Regression Testing
- Smoke Testing
- Integration Testing
- Data-driven Testing

Value For PLM Programs:

- Scalable QA strategy
- Cost & resources savings
- Comprehensive quality check
- New browser compatibility test
- Regression check on each delivery
- Reduced dependency on other resources
- Automated quality validation framework

Key components of SG-TAS:

- SG Web Recorder
- SG Native App Recorder
- SG-TAS Executor
- SG-TAS Reports
- SG Dataloader
- SG-TAS JIRA XRAY Integration



Created with the Personal Edition of HelpNDoc: [Streamline Your Documentation Process with HelpNDoc's Project Analyzer](#)

What's New

Features & Improvement:

- Native/Desktop Application Test Automation support (Native)
- SG Native/Desktop Application Recorder & Editor (Native)
- Chrome browser support apart from Firefox and IE (Web)
- Wait tag has capability to wait till current frame to loads with threshold time (Web)
- New feature added to mark list of tags as a warning tag. So that failure of the tag execute will not lead to failure of Test case. (Web/Native)
- New Feature added to execute test cases in browser's private/incognito mode (Web)
- JIRA XRAY Integration (Web/Native)
- Reporting Module bug fixing and enhancement (Web/Native)
- Web recorder bug fixing (Web)
- 3DX Data-loader improvement (Web)
- New Tag AssertPageLoadTime to assert the page loading time/to measure the client side page performance (Web)
- New Tag SelectColor to select the color from HTML5 Input type="color" (Web)
- New Tag DragAndDrop (Web)
- New Tag ELSEIF apart from IF...ELSE...ENDIF (Web)
- New Tag SelectSecurityContext added to select the security context from top user profile menu.(Web)
- Modified Tag OpenURL to open the URL in new tab and new window apart from same window/tab.(Web)

- Modified Tag InputText to enter encrypted text (Web)
- Browsers drivers download Automatically.(Web)
- Drivers instances Killing by executing .Bat File.(Web)
- SG TAS can perform automation without Config properties.
- Product License name renamed from "Selenium Test Automation Tool" to "**SG-TAS**" (Web/Native)

Created with the Personal Edition of HelpNDoc: [5 Reasons Why a Help Authoring Tool is Better than Microsoft Word for Documentation](#)

TAS-Prerequisite

- Web SG-TAS **prerequisite** :

1. First take Permission of Project Git clone from the Project Lead.
2. JDK version required for SG-TAS-2.1 :jdk1.8.0_251 or above of for 1.8.To specify a specific JDK when multiple are installed, set the "JAVA_HOME" environment variable to the directory path of the required JDK.
3. While Configuring the project with Eclipse make sure choose the gradle wrapper.
3. Make sure Machine should be up to Date.
4. OS : Windows 10 or other.
5. Use some BB for common functionalities tags or scenarios
6. Add some waits for loading the Web Elements on for locating.

- MQL Assert tag **prerequisite** :

1. Enovia folder's jars should be sync with latest 3DX stack version. :
 1. eMatrixServletRMI.jar
 2. enoviaKernel.jar
2. Basic understanding of MQL Query:

- Native TAS Recorder running **prerequisite** :

1. Screen regulations should be 100% and Display regulations will be 1920 * 1080
2. Java 8 should be installed on particular machine
3. Images should be proper and add some relation to that specific object
4. Common images should be identifiable on each devices
5. Use some BB for common functionalities On UI
6. Add some waits for loading the images

Created with the Personal Edition of HelpNDoc: [Effortlessly Create High-Quality Documentation with a Help Authoring Tool](#)

Get Started

Created with the Personal Edition of HelpNDoc: [Transform Your Word Doc into a Professional-Quality eBook with HelpNDoc](#)

Installer

3DX-TAS Portable Installation:

Following are the steps mentioned for installing package into your system

1. Here is how SG-TAS-2.1 folder looks like after extraction.

config	14-10-2024 17:45	File folder
└ data-loader	22-05-2024 16:41	File folder
└ dependencies	14-10-2024 17:59	File folder
└ documents	08-10-2024 03:42	File folder
└ drivers	14-10-2024 17:40	File folder
└ hub	24-09-2024 00:54	File folder
└ lib	14-10-2024 17:59	File folder
└ license	13-09-2024 00:50	File folder
└ logs	14-10-2024 17:16	File folder
└ node	24-09-2024 00:54	File folder
└ reports	22-05-2024 16:41	File folder
└ resources	24-09-2024 00:54	File folder
└ test-output	14-10-2024 17:16	File folder
└ testsuites	14-10-2024 17:16	File folder
└ web-recorder	22-05-2024 16:42	File folder
EncryptPassword.bat	22-05-2024 16:41	Windows Batch File 1 KB
EncryptPassword.sh	24-09-2024 00:54	SH Source File 1 KB
End_SG-TAS_Execution.bat	13-09-2024 00:50	Windows Batch File 1 KB
Execute_SG_TAS.bat	14-10-2024 17:40	Windows Batch File 1 KB
Execute_SG_TAS.sh	24-09-2024 00:54	SH Source File 1 KB
Execute_SG_TAS_NativeRecorder.bat	24-09-2024 00:54	Windows Batch File 1 KB
Execute_SG_TAS_NativeRecorder.sh	24-09-2024 00:54	SH Source File 1 KB
LogFileCompressor_Windows.bat	13-09-2024 00:52	Windows Batch File 1 KB
Stop_SG_TAS.bat	22-05-2024 16:41	Windows Batch File 1 KB

2. Following changes in **config** properties will get start you with default settings.
This folder contains configuration files.

- SG_TAS.properties
- Browser.properties
- Environment.properties
- Log4j.properties
- XRAY.properties

The above mentioned properties file is used to set value for different attributes(attributes values are set at both framework/application level depending upon the attribute used) . Below is the list of attributes, which needs to be set, as soon as the installation is complete(i.e. as pre-requisites) from different properties files:

▪ **SG_TAS.properties**

- **sg-tas.execution.title:** Execution server name. It is used to distinguish between different server execution and only used while reporting.
- **sg-tas.suitefile:** This key defines the suite file used for execution. Execution will be started based on test cases configured in this suite file. This is entry point of execution. User can combine multiple suite files in single master suite that we will call it as Test Plan/Test Lab. User need to give file path of this master suite file.

E.g.

testng.suitefile=Master.xml

The content of this suite file is shown in below image. User can add any number of child suites.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">
<suite name="Master">
    <suite-files>
        <suite-file path="testsuites\\WP01\\WP01.xml"></suite-file>
        <suite-file path="testsuites\\WP02\\WP02.xml"></suite-file>
    </suite-files>
</suite>
```

▪ **Browser.properties:**

- **sg-tas.browser.name:** Name of Browser: internetexplorer, firefox, chrome,edge and edge legacy
- **sg-tas.browser.webdriver.path:** User needs to provide the path of Web driver(i.e. executable file) for one of the browsers mentioned above,which would be present already in Driver folder
E.g: drivers/IEDriverServer_3.14.0_x32.exe, msedgedriver.exe

▪ **Enovia.Properties:**

- **sg-tas.environment.release:** This define Environment release level. This key required as based on release level, test cases will be executed.
Possible values for this key are *v6r2012x, v6r2013x, v6r2015x, v6r2017x, v6r2019x, v6r2020x, v6r2021x, v6r2022x, v6r2022xfd02, v6r2022xfd06, v6r2024xfd04, aras12, arasv28, arasv12sp9*
- **sg-tas.environment.url :** This key defines Environment URL to use for execution.
E.g. <https://www.steepgraphdev.com/3dspace>
- **sg-tas.3dspace.url.iscas :** If URL defines above key is CAS server URL then set this as **true** else **false**.

Created with the Personal Edition of HelpNDoc: [HelpNDoc's Project Analyzer: Incredible documentation assistant](#)

License

How To Get License For Test Automation Suite:

1. Double click on the application "LicenseRequestManager" to run.

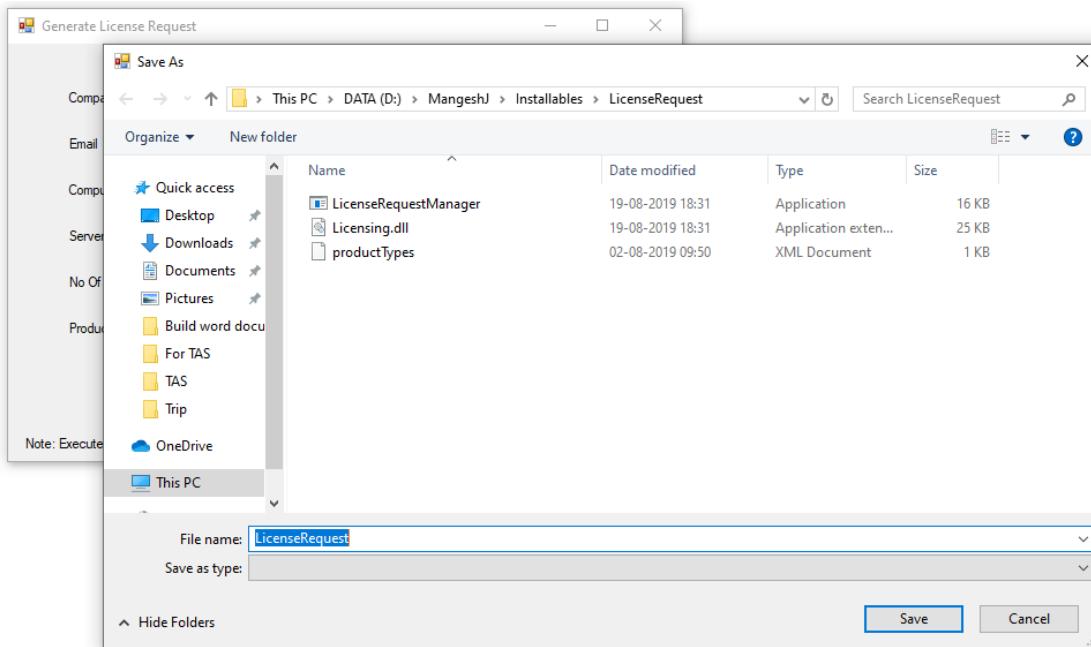
Name	Date modi...	Type
LicenseRequestManager	19-08-201...	Application
Licensing.dll	19-08-201...	Application exten...
productTypes	02-08-201...	XML Document

2. Enter the all the required details and select Product Type as Selenium Test Automation Tool

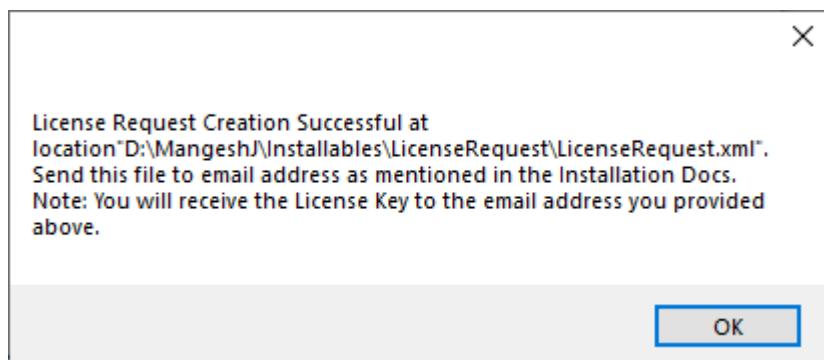
Generate License Request

Company Name	Your Company Name
Email	youremailid@company.com
Computer Name	<AUTO SELECTED>
Server Type	Production
No Of Users	1
Product Type	3DX-TAS
<input type="button" value="Generate Request File"/> <input type="button" value="Cancel"/>	
Note: Execute this binary on the system that will be hosting the Project	

3. Click on Generate Request File
 4. By default the Requested File will get save in same folder, but you can click the Browse button to select an alternative location.



5. Click on Save
6. Successful message will get displayed.



7. LicenseRequest file will get created on path mentioned in step 4, in xml format.
8. As mentioned in above message, send this file to SteepGraph support team
9. On behalf of SteepGraph support team, you will Receive one file "GeneratedLicense" in xml format.
10. It's only one machine.
11. Copy and Replace the file **./license/license.xml** in 3DX-TAS folder with new file that you have received. Make sure new file is renamed to **license.xml**.

Created with the Personal Edition of HelpNDoc: [Full-featured multi-format Help generator](#)

Property files

This **config** folder contains following property files.

- SG_TAS.properties
- Browser.properties
- Environment.properties
- Log4j.properties
- XRAY.properties

SG_TAS.properties

This configuration file defines all the necessary property attributes required for Test Execution Configuration.

Let's see the details of all the key(parameter) defined in this property file.

1. sg-tas.execution.title:

Execution server name. It is used to distinguish between different server execution and only used while reporting.

2. sg-tas.filter.suites:

In this, user needs to mention names of test suite, he/she wants to execute. If it is set as blank, then all test suites configured in xml file will be executed. This is just filter for test suite execution.

3. sg-tas.filter.testcases:

Similar to above key, this is just filter for test case execution. You should mention test case id here.

4. execution.loadscriptsonly:

Set this as true, if you want to load script in database only. Test cases will not be executed, if it is set as true.

5. sg-tas.suitefile:

This key defines the suite file, which is being used for execution. Execution will be started based on test cases configured in this suite file. This is entry point of execution. User can combine multiple suite files in single master suite, so that we will call it as Test Plan/Test Lab. User need to give file path of this master suite file.

E.g.

`sg-tas.suitefile=testsuites\|Master.xml`

The content of this suite file is shown in below image. User can add any number of child suites.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">
<suite name="Master">
    <suite-files>
        <suite-file path="testsuites\\WP01\\WP01.xml"></suite-file>
        <suite-file path="testsuites\\WP02\\WP02.xml"></suite-file>
    </suite-files>
</suite>
```

6. Parallel Execution using suite of suites :

For parallel execution, use suite-of-suites file as Master suite file

- a. **sg-tas.parallel.execution.poolsize:** This provides the thread pool size for parallel execution.
- b. **sg-tas.suitefile=** This provides the suite file, to which it should refer to. For parallel execution, please refer to Parallel Execution Section.

7. Driver Interval Settings :

- a. **sg-tas.execution.step.interval:** This defines wait interval between two xml tags in seconds
- b. **sg-tas.execution.step.timeout:** selenium web driver timeout in second
- c. **sg-tas.execution.step.timeout.pollinginterval:** driver polling interval

8. sg-tas.execution.step.retry.count:

Retry interval for failed xml tag

9. sg-tas.webservice.timeout:

This is used as timeout for web service request, hence should be used while automating the web service.

10. sg-tas.db.rerun.execution:

It has two possible values

- a. **false**: When you want to start new session.
- b. **True**: When you want to re-rerun failed test cases of existing session.

11. sg-tas.db.rerun.execution.criteria:

You should provide the criteria for re-execution of test cases from existing session. This is sql consisting of where clause, to filter the test cases for re-execution. User should provide the criteria based on execution status, in most cases.

Please find the database identifier for different status below:

- **1 - success**
- **0 - failed**
- **2 - ready**

e.g. - *execution.rerun.criteria=status not in (1)*

12. sg-tas.db.rerun.execution.date.format:

This key defines the date format of execution date (date provided in execution.date key). This date format is used for parsing the given date.

e.g.

*execution.date.format=DD-MM-YYYY
OR
execution.date.format=DD-MM-YYYY HH:MM:SS AM*

13. sg-tas.db.rerun.execution.date:

This is execution date of already executed session. This date is used to find the existing session from database. This will be used, if execution.rerun key is set to true.

e.g.

*execution.date=11-03-2019
OR
execution.date=11-03-2019 01:10:00 PM*

14. Database details to store Execution Status :

- a. **sg-tas.db.recording.enable**: This value is given as either **true** or **false** if its value is true, execution will be recorded into database else it will not recorded.
- b. **sg-tas.db.driver**: oracle.jdbc.driver.OracleDriver is the oracle's predefined driver in order to run SQL queries for fetching data.
- c. **sg-tas.db.jdbcurl**: This will be used as the jdbc url, in order to connect database.
e.g. *jdbc:oracle:thin:@192.168.229.128:1521:orcl*
- d. **sg-tas.db.username**: This is oracle user name, which is used to connect database.
- e. **sg-tas.db.password**: This is oracle user's password. It is mandatory to provide password in encrypted format. Encrypted password can be generated using TAS password encryption utility.
- f. **sg-tas.db.acquireincrement**: Since Connection acquisition is very slow, it is almost always useful to increase the number of Connections eagerly, in batches, rather than forcing each client to wait for a new Connection to provoke a single acquisition when the load is increasing. acquireIncrement determines how many Connections a c3p0(connection) pool will attempt to acquire when the pool has run out of Connections. Regardless of acquireIncrement, the pool will never allow maxPoolSize to be exceeded. The number of Connections in a pool decreases whenever a pool tests a Connection and finds it to be broken , or when a Connection is expired by the

- pool after sitting idle for a period or for being too old.
- g. **sg-tas.db.initialpoolsize:** It is pre-defined pool size.
 - h. **sg-tas.db.maxpoolsize:** It is maximum pool size.
 - i. **sg-tas.db.minpoolsize:** It is minimum pool size.
 - j. **sg-tas.db.maxstatements:** maxStatements is JDBC's standard parameter for controlling statement pooling. maxStatements defines the total number Prepared Statements a Data Source will cache. The pool will destroy the least-recently-used PreparedStatement when it hits this limit. This sounds simple, but it's a strange approach, because cached statements conceptually belong to individual Connections; they are not global resources. To figure out a size for maxStatements that does not "churn" cached statements, you need to consider the number of frequently used PreparedStatements in your application and multiply that by the number of Connections you expect in the pool (maxPoolSize in a busy application).

15. Execution E-Mail Notification :

- a. **sg-tas.mail.sendreport.enable:** It has values either **True** or **False**. If it's True, then report will be sent to the person which is included in mail.from.
- b. **sg-tas.mail.from:** This includes the id of the person, from whom mail has been sent.
- c. **sg-tas.mail.to:** This includes the id of that person, to whom mail has been sent.
- d. **sg-tas.mail.cc:** This includes id of the person, who needs to be in cc.
- e. **sg-tas.mail.bcc:** This includes id of the person, who needs to be in bcc.
- f. **sg-tas.mail.smtp.username:** This includes smtp server user name.
- g. **sg-tas.mail.smtp.password:** This includes smtp server user password.
- h. **sg-tas.mail.smtp.host:** This includes smtp server's hostname
- i. **sg-tas.mail.smtp.port:** This includes smtp server's port number
- j. **sg-tas.mail.subject:** It contains mail subject.
- k. **sg-tas.mail.body:** It contains content of the mail.

16. Failed Test Case Configuration :

- a. **sg-tas.snapshot.enable:** Set it as true, if user want to capture snapshot of failed test cases, else set it as false.
- b. **sg-tas.video.recording.enable:** Set it as true, if you want to capture video of test case execution, else set it as false.

Usability Configuration :

17. **sg-tas.system.notification.enable:** It has two values either **True** or **False**. If its value is true, then system notification will be enabled.
18. **sg-tas.highlight.webelement.enable:** Set it as true, if you want to highlight element on web page, before doing given action or else set it as false.
19. **sg-tas.highlight.webelement.style:** Here you can specify css for highlighting the element on web page. If you don't specify any css, the default css will be applied.
e.g. `highlight.style= border: 2px solid rgba(81, 203, 238, 1);padding: 2px;`
20. **sg-tas.testresults.to.outputfolder:** With the help of this key, user can redirect test results to ./test-output/test-results folder. It is considered as a true, by default.
21. **sg-tas.testresults.overwrite:** With the help of this key, user can keep previous execution results by setting it to false, new folder will be created with time-stamp
22. **sg-tas.skiperror.taglist:** With the help of this tag, user could specify tags to ignore the error during the execution. If there are multiple tags for which user want to ignore the error, user could separate them by "comma". This field should be left blank, in case no tags are applicable for this [3dx-tas.skiperror.taglist=AssertPageLoadTime,OpenActionToolbarMenu]

Other Configuration :

23. **sg-tas.execution.private.key:** This key enables us to choose different private key for

encryption/decryption. This can be defined by user and must not be empty.

Browser.properties:

1. **sg-tas.browser.name** : Name of Browser: internetexplorer, firefox, chrome, edge, edge legacy
2. **sg-tas.browser.webdriver.path**: This is the path of driver with respect to browser name. All the latest drivers will be present in drivers folder
3. **sg-tas.browser.headless** : If user want to run script in background, then set value of this key to true, else false. Headless browser has lot of limitation from selenium, so we do not recommend this option.
4. **sg-tas.browser.firefox.marionette** : This key is specific to Firefox browser. If user wants to use Firefox version less than 47, then set value of this key to true else, it should false.
5. **sg-tas.browser.ie.requirefocus**: This key is specific to IE browser and if user want to move mouse cursor to web element before performing any action on it then set it as true else false. Recommended value of this key is false.
6. **sg-tas.browser.nativeevents** : This key is used to enable or disable native OS events. If user want to enable it, then set it as true else false. Recommended value of this key is true.
7. **sg-tas.browser.remote**: This key defines whether test cases to be executed on local machine or on remote client machine. If its value is false, then execution is on local machine else it is on remote machine.
8. **sg-tas.browser.privatemode**: This property enables users to launch browser and run scripts in Private Mode. By default, this property value is false. If user wants to enable private mode, then he/she needs to set value of this key as true.
9. **sg-tas.browser.remote.platform**: This key will be used when browser.remote key is set to true. Its value should be taken from grid console page. Please check [Remote Execution](#) section for more details.
10. **sg-tas.browser.remote.nodeurl** : This is the URL of remote machine, where node will be configured for test cases execution.
e.g. <http://192.168.1.27:5566/wd/hub>
11. **sg-tas.browser.proxy** : Default value of this is false. Setting this parameter value to true indicates that browser proxy will be used for entire test plan being executed. For configuring and calculating performance logs, we need to set value of this parameter to true. If this parameter is false and need to execute certain number of test cases in test plan with browser proxy then use the 'proxy' parameter eg `<parameter name="proxy" value="true"></parameter>` inside the `<test> </test>`.
12. **sg-tas.browser.acceptInsecureCerts=true** : This key used for When there is no valid certificate available for https connection, browser will show this link(certificate error) while opening new window.
13. **sg-tas.browser.date.format** : This key is used when user want to print current system date/validated the product created date with the current system date. The default value for this key is in format [dd/MM/yyyy]. User can change the date format as per the requirement.

Environment.properties

1. **sg-tas.environment.release** : This defines Enovia release level. This key is used for release level, Based on release levels, the test cases will be executed. Possible values for this key as follow:

- *v6r2012x, v6r2013x, v6r2015x, v6r2017x, v6r2019x, v6r2020x, v6r2021x, v6r2022x, v6r2022xfd02, v6r2022xfd06, v6r2024xfd04, aras12, arasv28, arasv12sp9*
2. **sg-tas.environment.url** : This key defines Enovia/Aras URL (url of the environment which is going to be automated) to be used for execution.
E.g. <https://www.steepgraphdev.com/3dspace>
 3. **sg-tas.3dspace.url.iscas** : If URL defines above key, as isCAS server URL, then set this as either **true** else **false**.
 4. **sg-tas.3dspace.url.verify**: This key defines whether user need to validate enovia url before execution or not. If it's true, then it would validate the enovia URL, else not.
 5. **sg-tas.3dspace.has.tvc**: Set it as true when TVC framework will be installed on Enovia, else set it as false.
 6. **sg-tas.3dspace.login.has.securitycontextpage**: If Enovia is showing security context page after login, then set this as **true** or else set this as **false**.
 7. **sg-tas.3dspace.validation.url**: This is enovia internal server url, which is required for validation purpose.
 8. **sg-tas.3dspace.validation.username**: This key defines user name, which is used to connect to enovia while doing the validation. User must be enovia system administrator having all rights.
 9. **sg-tas.3dspace.validation.password**: This should be used as the password of enovia, which user defines in enovia.validation.username key.

Log4j.properties

This is configuration file to generate application logs. All required configuration is already define, no need to change it.

Following is the screenshot of Log4j.properties file:

```

#Root Logger
log4j.rootLogger=DEBUG,file
log4j.appender.file=org.apache.log4j.RollingFileAppender
log4j.appender.file.File=logs\\Selenium.log
log4j.appender.file.maxFileSize=50MB
log4j.appender.file.maxBackupIndex=1000
log4j.appender.file.layout=org.apache.log4j.PatternLayout
log4j.appender.file.layout.ConversionPattern=%d [%c] %x - %m%n
log4j.appender.file.Append=false

#Application Debug Logger
log4j.logger.SG-TAS=DEBUG, TestDebug
log4j.appender.TestDebug=org.apache.log4j.RollingFileAppender
log4j.appender.TestDebug.maxFileSize=50MB
log4j.appender.TestDebug.maxBackupIndex=1000
log4j.appender.TestDebug.Threshold=DEBUG
log4j.appender.TestDebug.layout=org.apache.log4j.PatternLayout
log4j.appender.TestDebug.layout.ConversionPattern=%d [%c] %x - %m%n
log4j.appender.TestDebug.File=logs\\TestAutomationDebug.log
log4j.appender.TestDebug.Append=false

#Application Error Logger
log4j.logger.SG-TAS-ERROR=ERROR, TestError
log4j.appender.TestError=org.apache.log4j.RollingFileAppender
log4j.appender.TestError.maxFileSize=10MB
log4j.appender.TestError.maxBackupIndex=1000
log4j.appender.TestError.Threshold=ERROR
log4j.appender.TestError.layout=org.apache.log4j.PatternLayout
log4j.appender.TestError.layout.ConversionPattern=%d [%c] %x - %m%n
log4j.appender.TestError.File=logs\\TestAutomationError.log
log4j.appender.TestError.Append=false

#To Record Execution Result in separate File
log4j.logger.SG-TAS-RESULT=DEBUG, RecordDebug
log4j.appender.RecordDebug=org.apache.log4j.RollingFileAppender
log4j.appender.RecordDebug.maxFileSize=10MB
log4j.appender.RecordDebug.maxBackupIndex=1000
log4j.appender.RecordDebug.Threshold=DEBUG
log4j.appender.RecordDebug.layout=org.apache.log4j.PatternLayout
log4j.appender.RecordDebug.layout.ConversionPattern=%m%n
log4j.appender.RecordDebug.File=test-output\\SG-TAS_ExecutionResult.csv
log4j.appender.RecordDebug.Append=false

#To Record Performance Logs in separate File
log4j.logger.SG-TAS-PERFORMANCE=PERFORMANCE, RecordPerformance
log4j.appender.RecordPerformance=org.apache.log4j.RollingFileAppender
log4j.appender.RecordPerformance.maxFileSize=50MB
log4j.appender.RecordPerformance.maxBackupIndex=1000
log4j.appender.RecordPerformance.Threshold=PERFORMANCE
log4j.appender.RecordPerformance.layout=org.apache.log4j.PatternLayout
log4j.appender.RecordPerformance.layout.ConversionPattern=%d [%c] %x - %m%n
log4j.appender.RecordPerformance.File=logs\\Performance.log
log4j.appender.RecordPerformance.Append=false

#To Record Execution Result in separate File
log4j.logger.SG-TAS-PERFORMANCE-RESULT=DEBUG, RecordPerformanceDebug
log4j.appender.RecordPerformanceDebug=org.apache.log4j.RollingFileAppender
log4j.appender.RecordPerformanceDebug.maxFileSize=10MB
log4j.appender.RecordPerformanceDebug.maxBackupIndex=1000
log4j.appender.RecordPerformanceDebug.Threshold=DEBUG
log4j.appender.RecordPerformanceDebug.layout=org.apache.log4j.PatternLayout
log4j.appender.RecordPerformanceDebug.layout.ConversionPattern=%m%n
log4j.appender.RecordPerformanceDebug.File=${perf_result.csv}
log4j.appender.RecordPerformanceDebug.Append=false

```

XRAY.properties

1. **jira.xray.use.integration** - This value is given as either true or false. If true, execution will be started for Jira XRay module else it will execute TAS normally without the module taken into consideration.
2. **jira.xray.testsuites.folders** - In this key, user has to mention name of folder(s) located in

testsuites folder, he/she wants to execute.

For ex. LH0_3DXPlatform - Multiple values can also be provided in a be comma separated format without spaces between the different names and commas.

For ex. LH0_3DXPlatform,ERMATT

3. **folder.jira.xray.object.execute** - This key specifies the Test Case/Test Plan/Story number for execution. There has to be an issue in JIRA corresponding to the test case/test plan/story with the JIRA issue's number matching the names in TAS module. The 'folder' in the key must be replaced by the name of the actual folder in the testsuites folder. There can be multiple test cases/test plans/stories and there can be multiple instances of this key accordingly.

For ex.

LH0_3DXPlatform.jira.xray.object.execute = PLM2019-4861 (PLM2019-4861 corresponds to the issue number in JIRA and the folder contains the test case files.)

LH0_3DXPlatform.jira.xray.object.execute = PLM2019-5192

ERMATT.jira.xray.object.execute = PLMRM2019-404

4. **jira.xray.test.environment** - This is a name given to the environment on which test cases will be executed. This can be any string. ex. steepgraphdev, plmint01, etc.

5. **jira.xray.url** - This key defines the JIRA instance URL to be used for execution.

E.g. https://eteamproject.internal.ericsson.com

6. **jira.xray.port** - This key defines JIRA server's port number

E.g. - jira.xray.port = 443

7. **jira.xray.username** - This key defines user name, which is used to connect to Jira while testing.

E.g.- jira.xray.username = zjagras

8. **jira.xray.password** - This should be the password for the username defined in the key jira.xray.username. The password is encrypted.

E.g.- jira.xray.password = 9m+MP0Y9NBQTanUZRZmU7Q==

9. **jira.xray.output.File** - With the help of this key, user can keep execution results in a predefined csv file.

E.g.- jira.xray.output.File=test-output/SG-TAS_ExecutionResult.csv

10. **jira.xray.testExecution.promotion** - This key defines the id of the workflow transition state to which the test execution object created for a test case execution should be promoted post execution irrespective of it's success/failure. It is obtained by using the jira rest hook: http://localhost:8080/rest/api/2/issue/<test_execution_jira_number>/transitions?expand=transitions.fields where <test_execution_jira_number> is the test execution object's number.

Steps to Download and Analyze the XML Workflow File:

1. Verify Admin Access

- Ensure you have administrator privileges to access and download the workflow XML file.

2. Modify and Access the URL

- Use the following URL format to download the XML file:



- Replace `http://localhost:8080` with your domain ID.
- Replace `TAS` in `TAS+Workflow` with the name of your project.

URL :

http://localhost:8080/secure/admin/workflows/ViewWorkflowXml.jspa?workflowMode=

spa?workflowMode=live&workflowName=TAS+Workflow

3. Log In

- Open the URL in your browser.
- Log in using your administrator credentials.

4. Download the XML File

- Upon successful login, the browser will download the XML file corresponding to the specified workflow.

5. Review the XML File Structure

- The XML file structure will resemble the provided example, with key sections such as `<workflow>`, `<initial-actions>`, and `<common-actions>`.

```


<common-actions>
    <action id="2" name="Close Issue" view="resloveissue">
        <action id="3" name="Reopen Issue" view="commentassign">
            <action id="4" name="Start Progress">
                <action id="5" name="Resolve Issue" view="resloveissue">


```

Key Information in the XML File:

Common-Actions Section

- Each `<action>` tag defines a transition state.
 - Attributes:
 - `id` : Represents the transition state number.
 - `name` : Represents the transition state name.
- Example:

xml

```

<action id="2" name="Close Issue" view="resloveissue">
    <!-- Additional meta and conditions -->
</action>

```

- The `<unconditional-result>` tag within an `<action>` defines the resulting state after the transition.

So here the jira.xray.testExecution.promotion are 2, 3, 4, 5.

11. For ex. The url: <http://localhost:8080/rest/api/2/issue/TES-50/transitions?expand=transitions.fields> gives the output in JSON format as below. If a user wants the test execution object to be promoted to Done state after execution, the id from the below json output for Done state is 41. This should be used in the key as :
jira.xray.testExecution.promotion=41

```

{
  "expand": "transitions",
  "transitions": [
    {
      "id": "11",
      "name": "To Do",
      "to": {
        "self": "http://localhost:8080/rest/api/2/status/10000",
        "description": "",
        "iconUrl": "http://localhost:8080/",
        "name": "To Do",
        "id": "10000",
        "statusCategory": {

```

```

"self":"http://localhost:8080/rest/api/2/statuscategory/2",
"id":2,
"key":"new",
"colorName":"blue-gray",
"name":"To Do"
},
{

"fields":{

},
{
"id":21,
"name":"In Progress",
"to":{

"self":"http://localhost:8080/rest/api/2/status/3",
"description":"This issue is being actively worked on at the moment by the assignee.",
"iconUrl":"http://localhost:8080/images/icons/statuses/inprogress.png",
"name":"In Progress",
"id":3,
"statusCategory":{

"self":"http://localhost:8080/rest/api/2/statuscategory/4",
"id":4,
"key":"indeterminate",
"colorName":"yellow",
"name":"In Progress"
}
},
"fields":{

},
{
"id":31,
"name":"In Review",
"to":{

"self":"http://localhost:8080/rest/api/2/status/10001",
"description":"",
"iconUrl":"http://localhost:8080/",
"name":"In Review",
"id":10001,
"statusCategory":{

"self":"http://localhost:8080/rest/api/2/statuscategory/4",
"id":4,
"key":"indeterminate",
"colorName":"yellow",
"name":"In Progress"
}
},
"fields":{

},
{
"id":41,

```

```

"name":"Done",
"to":{
  "self":"http://localhost:8080/rest/api/2/status/10002",
  "description":"",
  "iconUrl":"http://localhost:8080/",
  "name ":"Done",
  "id":"10002",
  "statusCategory":{
    "self":"http://localhost:8080/rest/api/2/statuscategory/3",
    "id":3,
    "key":"done",
    "colorName":"green",
    "name":"Done"
  }
},
"fields":{

}
]
}

```

12. **3dx-tas.video.recording.enable** - Set it to true, if you want to capture video of test execution, else false.
13. **jira.xray.video.upload** - This key uploads video on the jira only if it is set to true. The default value is false.

Created with the Personal Edition of HelpNDoc: [Easily create EBooks](#)

Reporting app WAR

Steps to Configure TAS Reporting:

We need to implement some steps using files present in **TestAutomationbuild** folder for proper working of TAS Reporting.

1. Install Oracle database on local machine. Make sure oracle home path is set in environment variables.
2. Open Oracle SQL Developer. Create connection and connect with database.
3. Create database for TestAutomation Reporting module using **SQL** script(**TestAutomationDB.sql**) present in **\resources\main**.
4. Set the below properties in **3DX_TAS.properties** placed at **\config**:
 - **sg-tas.db.recording.enable=true**
 - **sg-tas.db.driver=oracle.jdbc.driver.OracleDriver**
 - **sg-tas.db.jdbcurl=<jdbc_url>**
 - **sg-tas.db.username=<user_name>**
 - **sg-tas.db.password=<password>**

Encrypted password must be used in **3dx-tas.db.password** property field which can be generated using TAS password encryption utility.

5. Deploy the war file **SGS3DXTASReport.war** present at **.reports** folder in Apache Tomcat server by placing the war file in **webapps** folder under **Tomcat** and start the **Tomcat** service.
6. Once deployed and extracted in **webapps** folder, stop the **tomcat** service. Modify the

SGS3DXTASReport\WEB-INF\data-source-cfg.xml file to have correct database configuration as in Step 1 and 2.

7. Start **tomcat** again and browse the **url.(http://localhost:8080/SGS3DXTASReport)**

8. To start the **apache tomcat** service:

- Start a Command Prompt from the Start menu.
- Navigate to the Tomcat bin directory, e.g., cd c:\Tomcat8\bin.
- Type in startup and then hit Enter to execute the Tomcat server start up script.
- To stop the Tomcat server, type in shutdown and then hit Enter in the original command prompt.

Created with the Personal Edition of HelpNDoc: [Benefits of a Help Authoring Tool](#)

Default Property

Default Properties

In TAS we have define default property keys so that without property files , TAS can able to perform automation testing.

#3DX_TAS.properties

```

#Execution server name. It is only used in reporting to distinguish different
server execution
sg-tas.execution.title=QA-Run-01

#Master Suite File Path.
sg-tas.suitefile=TestPlan.xml

#Name of Test Suites to execute
sg-tas.filter.suites=

#Test Cases Id to execute
sg-tas.filter.testcases=

#This property will be taken in account only if sg-tas.3dspace.url is blank or not
present
sg-tas.initial.url=
sg-tas.initial.url.verify=false

#For parallel execution of suites or test cases. set value more than 1. Default
value is 1, means no parallel execution.
sg-tas.parallel.execution.poolsize=1

#This define wait interval between two xml tags in second
sg-tas.execution.step.interval=0

#selenium web driver timeout in second
sg-tas.execution.step.timeout = 30
sg-tas.execution.step.timeout.pollinginterval = 5

#retry count for failed xml tag
sg-tas.execution.step.retry.count=2

#This key is private key used for encryption/decryption and should not be blank
sg-tas.execution.private.key=Bar12345Bar12345

#WebService

```

```

sg-tas.webservice.timeout=

#With this key user can redirect test results to ./test-output/test-results
folder, by default it is considered as a true
#true
#false : on false results will be created in corresponding test script folder.
sg-tas.testresults.to.outputfolder=true

#With this key user can keep previous execution results by setting it to false,
new folder will be created with time-stamp
#Default Value : true
#Possible Values : true,false
sg-tas.testresults.overwrite=false

#Failure Test Case Snapshot
# Snapshot will be recorded in output folder based on key: 3dx-
tas.testresults.to.outputfolder
sg-tas.snapshot.enable=false

#Test Case Video Recording
#Video will be recorded in output folder based on key: 3dx-
tas.testresults.to.outputfolder
sg-tas.video.recording.enable=true

#Test Case Video Format
#Default format is mov, other possible value mp4
sg-tas.video.format=mp4

# Usability Configuration
-----
#show notification on windows system tray
sg-tas.system.notification.enable=true

#Highlight Element on Web Page
sg-tas.highlight.webelement.enable=true

#This key contains css that will be applied to web element
#box-shadow: 0 0 10px rgba(81, 203, 238, 1);border: 2px solid rgba(81, 203, 238,
1);padding: 2px;
sg-tas.highlight.webelement.style=

#Specify tags to ignore the error during the execution.
#Comma separated list of tags. Leave it blank if no tags are applicable for
skiperror
sg-tas.skiperror.taglist=

# Database details to store Execution Status
-----
# to record test case execution State
sg-tas.db.recording.enable=false

sg-tas.db.driverclass=oracle.jdbc.driver.OracleDriver
sg-tas.db.jdbcurl=jdbc:oracle:thin:@192.168.0.189:1521:R2019x
sg-tas.db.username=SeleniumTest19x
sg-tas.db.password=SeleniumTest19x
sg-tas.db.acquireincrement=10
sg-tas.db.initialpoolsize=10
sg-tas.db.maxpoolsize=200
sg-tas.db.minpoolsize=10
sg-tas.db.maxstatements=200

#Load Script to database only

```

```

#true -- only scripts loaded to database
#false -- script loaded to database and execution started immediately. This is
Default value
sg-tas.db.record.scriptsonly=false

# Failed Test Case Configuration
-----
# Re-RUN Execution
sg-tas.db.rerun.execution=false

#This is option. User can add extra filter while rerunning the execution.
#status
#1 -- success
#0 -- failed
#2 -- ready
#Default criteria is status not in (1)
sg-tas.db.rerun.execution.criteria=

#This date format could be used: DD-MM-YYYY OR DD-MM-YYYY HH:MI:SS AM
sg-tas.db.rerun.execution.date.format=DD-MM-YYYY

#This date should be given in above date fomat
sg-tas.db.rerun.execution.date=11-03-2019

# Execution E-Mail Notification
-----
sg-tas.mail.sendreport.enable=false
sg-tas.mail.from=amitn@steepgraph.com
sg-tas.mail.to=amitn@steepgraph.com
sg-tas.mail.cc=amitn@steepgraph.com
sg-tas.mail.bcc=amitn@steepgraph.com
sg-tas.mail.smtp.username=amitn@steepgraph.com
sg-tas.mail.smtp.password=
sg-tas.mail.smtp.host=smtp.gmail.com
sg-tas.mail.smtp.port=587
sg-tas.mail.subject=
sg-tas.mail.body=
```

Browser Details for Automation Testing

#Browser.properties

```

#Name of Browser: internetexplorer, firefox, chrome, edge, edgelegacy
#sg-tas.browser.name=internetexplorer
#sg-tas.browser.name=firefox
sg-tas.browser.name=chrome
#browser value for both Edge Legacy
#sg-tas.browser.name=edgelegacy
#browser value for Edge
#sg-tas.browser.name=edge
#sg-tas.browser.name=safari

#sg-tas.browser.webdriver.path=drivers/IEDriverServer_Win32_3.150.1.exe
#sg-tas.browser.webdriver.path=drivers/geckodriver.exe
#sg-tas.browser.webdriver.path=drivers/chromedriver1.exe
#driver for browser Edge Legacy
#sg-tas.browser.webdriver.path=drivers/MicrosoftWebDriver.exe
#driver for browser Edge
sg-tas.browser.webdriver.path=drivers/msedgedriver.exe
#sg-tas.browser.webdriver.path=/usr/bin/safaridriver
```

```

sg-tas.tag.openurl.url =https://outlook.office.com
sg-tas.browser.webdriver.autodownload=false
sg-tas.browser.proxy=false

sg-tas.browser.headless=false
sg-tas.browser.firefox.marionette=true
sg-tas.browser.ie.requirefocus=false
sg-tas.browser.nativeevents=true
#Private mode is considered only if sg-tas.browser.headless=false
#For IE Private mode : need to set the registry
HKEY_CURRENT_USER\Software\Microsoft\Internet Explorer>Main\TabProcGrowth to 0
sg-tas.browser.privatemode=false
sg-tas.browser.enableOTP = false
sg-tas.browser.enable.draganddropjs.generic = true

#Below keys used for remove execution for Web Driver
sg-tas.browser.remote=false
sg-tas.browser.remote.platform=WINDOWS
sg-tas.browser.remote.nodeurl=http://192.168.1.5:5566/wd/hub

```

#Environment.properties

```

#Enovia server details
-----
sg-tas.environment.release=V6R2021x
#sg-tas.environment.url=https://www.steepgraphdev21x.com/3dspace
sg-tas.3dspace.url=https://www.steepgraph2022x.com/3dspace
#sg-tas.3dspace.url=https://www.sgsdev22x.com/3dspace
#sg-tas.3dspace.url=https://www.steepgraph2021x.com/3dspace
sg-tas.3dspace.url.verify=false
sg-tas.3dspace.url.iscas=true
sg-tas.3dspace.has.tvc=false
sg-tas.3dspace.login.has.securitycontextpage=false

#This keys will be used to run mql
#This is URL must be internal server URL
#sg-tas.3dspace.validation.url=http://192.168.0.189:8070/internal
sg-tas.3dspace.validation.url=
sg-tas.3dspace.validation.username=creator
sg-tas.3dspace.validation.password=

```

XRAY.properties

```

#JIRA Xray and SG-TAS Integration
#Start Execution in Integration

jira.xray.use.integration=false

jira.xray.testsuites.folders =

# Test Case/Test Plan/Story Id
TASTestingProject.jira.xray.object.execute =

jira.xray.test.environment=UAT01

jira.xray.url = https://eteamproject.internal.ericsson.com
jira.xray.port = 443
jira.xray.username = zketwag
jira.xray.password = qFSeq8Ba2nKytTOISxydBw==

```

```

# Generate Output
jira.xray.output.File=test-output/sg-TAS_ExecutionResult.csv

#Update only by Admin
jira.xray.testExecution.promotion=91,11,41

#Upload video on the jira only if it is true.
#If upload is true then execution object will take lot of time to be created.
#This setting will work for the following TAS setting only:
#sg-tas.video.record.enable=true
#sg-tas.video.record.path=testsuites
#Note: For large Video Uploads to JIRA, change the JIRA setting by the
Administrator.
# Settings --> System --> Attachments --> Edit Settings
# Set Attachment Size = 50MB or any desired proper value.
jira.xray.video.upload=false

jira.xray.execute.selective=
jira.xray.execute.selective.status=
#possible values:todo,pass,fail,aborted

```

Created with the Personal Edition of HelpNDoc: [Free HTML Help documentation generator](#)

How To

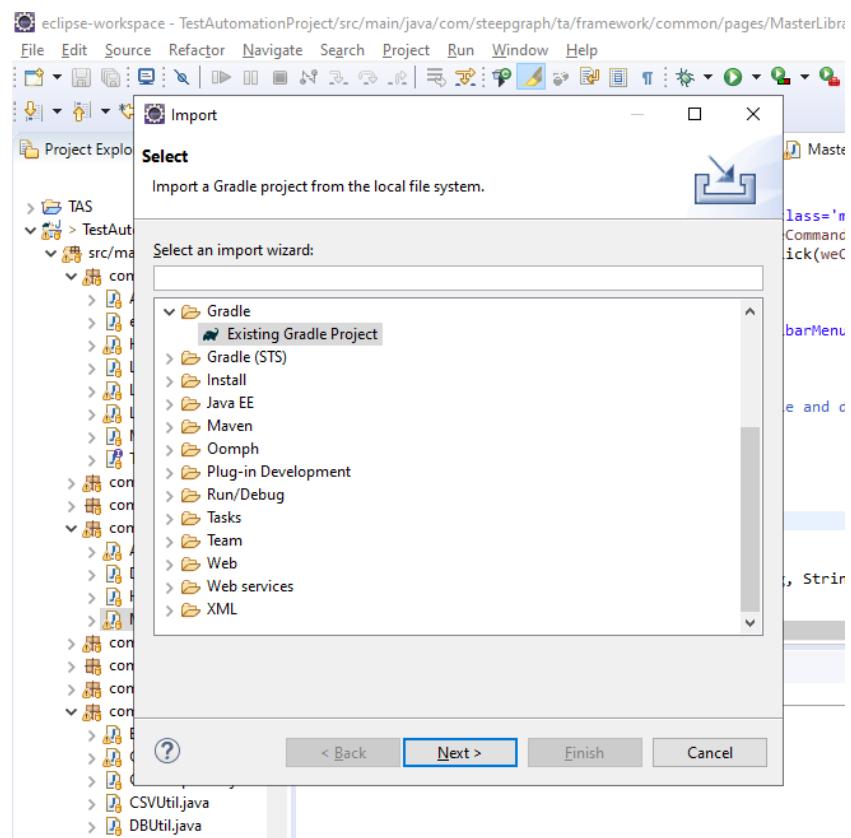
Created with the Personal Edition of HelpNDoc: [Experience a User-Friendly Interface with HelpNDoc's Documentation Tool](#)

Configure Eclipse

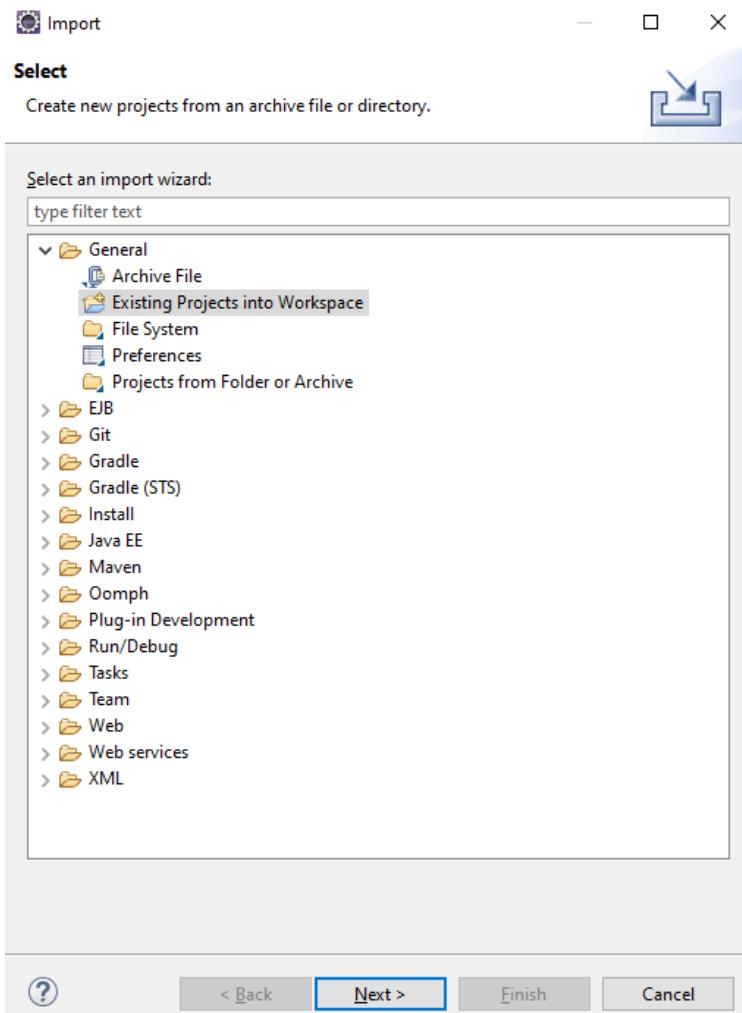
How to Configure 3DX-TAS on Eclipse

Following are the steps to configure framework project on eclipse :

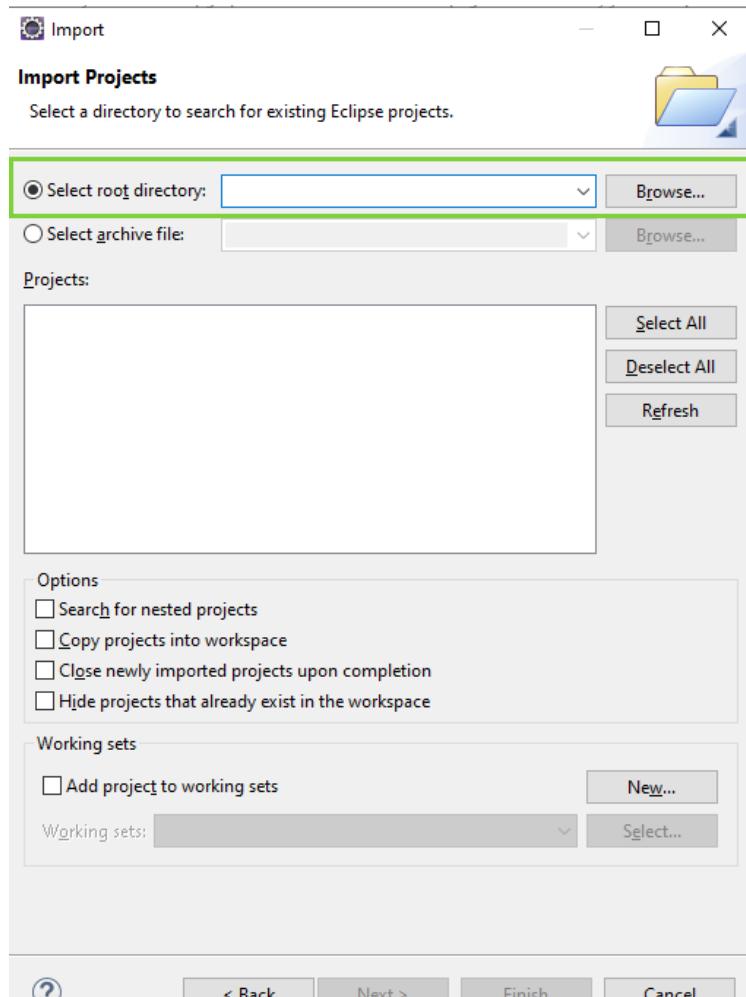
1. Open Eclipse.
2. From the main menu bar, select **File>Import....** The Import wizard opens.



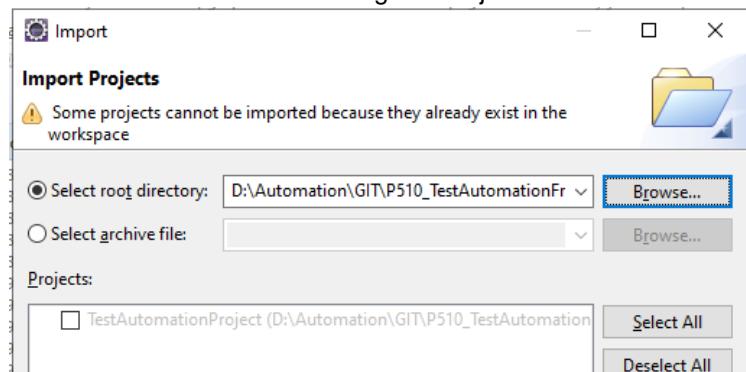
3. Select **General > Existing Project into Workspace** and click **Next**.



4. Choose **Select root directory** and click the associated **Browse** to locate the file containing the projects



5. Now Select the Folder containing the Project and Click on **Finish** to start the import.



6. Once project is imported verify folder structure of SG-TAS-201

 config	14-10-2024 17:45	File folder
 data-loader	22-05-2024 16:41	File folder
 dependencies	14-10-2024 17:59	File folder
 documents	08-10-2024 03:42	File folder
 drivers	14-10-2024 17:40	File folder
 hub	24-09-2024 00:54	File folder
 lib	14-10-2024 17:59	File folder
 license	13-09-2024 00:50	File folder
 logs	14-10-2024 17:16	File folder
 node	24-09-2024 00:54	File folder
 reports	22-05-2024 16:41	File folder
 resources	24-09-2024 00:54	File folder
 test-output	14-10-2024 17:16	File folder
 testsuites	14-10-2024 17:16	File folder
 web-recorder	22-05-2024 16:42	File folder
 EncryptPassword.bat	22-05-2024 16:41	Windows Batch File
 EncryptPassword.sh	24-09-2024 00:54	SH Source File
 End_SG-TAS_Execution.bat	13-09-2024 00:50	Windows Batch File
 Execute_SG_TAS.bat	14-10-2024 17:40	Windows Batch File
 Execute_SG_TAS.sh	24-09-2024 00:54	SH Source File
 Execute_SG_TAS_NativeRecorder.bat	24-09-2024 00:54	Windows Batch File
 Execute_SG_TAS_NativeRecorder.sh	24-09-2024 00:54	SH Source File
 LogFileCompressor_Windows.bat	13-09-2024 00:52	Windows Batch File
 Stop_SG_TAS.bat	22-05-2024 16:41	Windows Batch File

7. Configure the intelliSense for writing the test cases.

- Add new XML in testsuites folder.

- Copy below content to your test case xml.

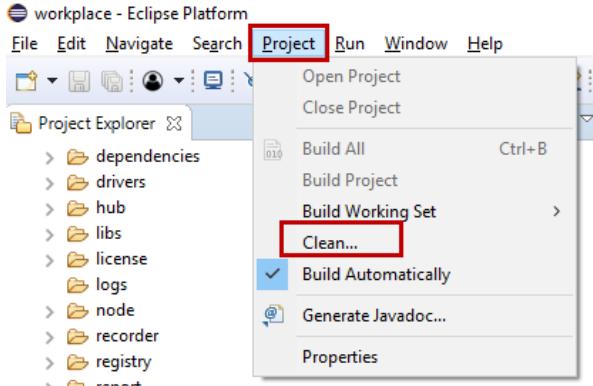
```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<TestCase xmlns="https://www.steepgraph.com"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="https://www.steepgraph.com ../../
  ..../resources/xsd/TestAutomationFramework.xsd">
```

`</TestCase>`

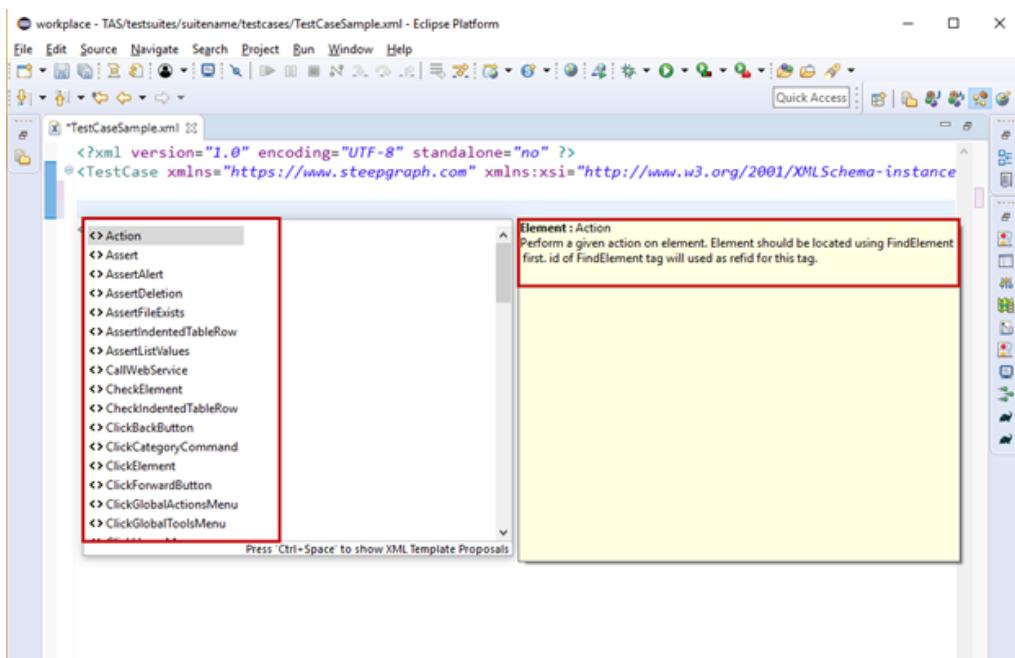
Update xsi:schemalocation attribute of TestCase tag to location of your TestAutomationFramework.xsd. This file should be present in resources/main folder of your build.

- Give relative as shown above or copy this xsd to same folder where test case xml is and just give xsd name.

- Close test case xml and clean the project.



- Open the test case xml. Press keyboard ctrl+space keys, you will get intelliSense.



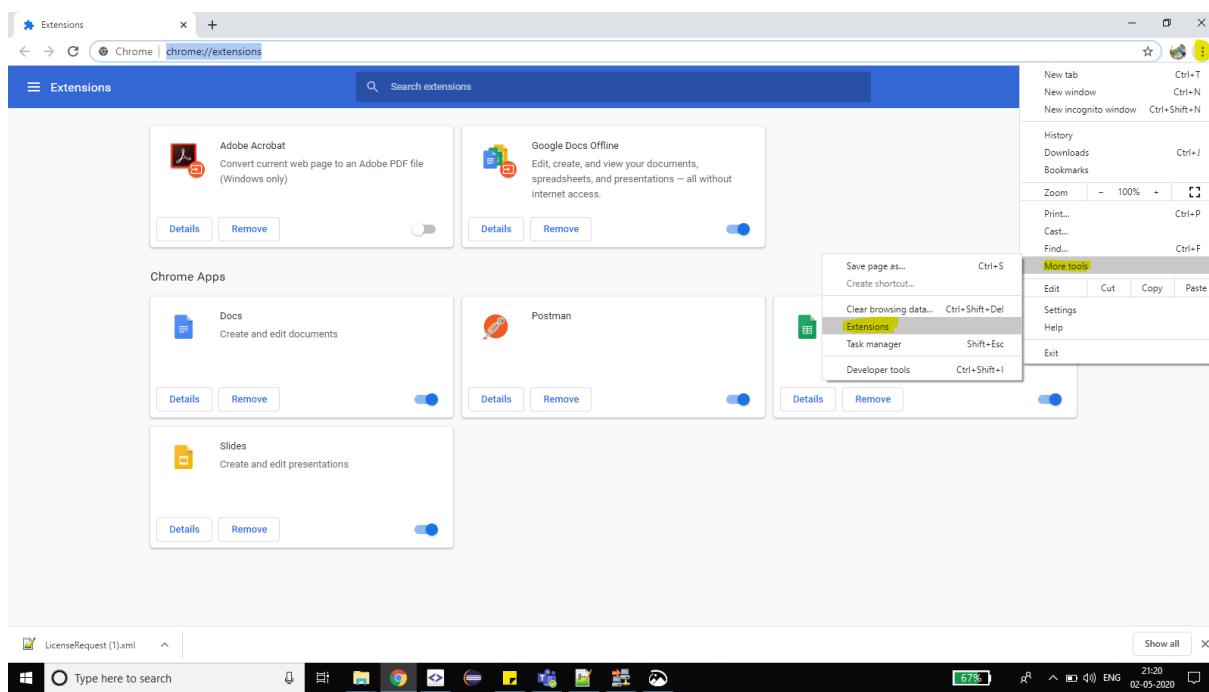
Created with the Personal Edition of HelpNDoc: [Transform Your Documentation Process with HelpNDoc's Project Analyzer](#)

Web Recorder

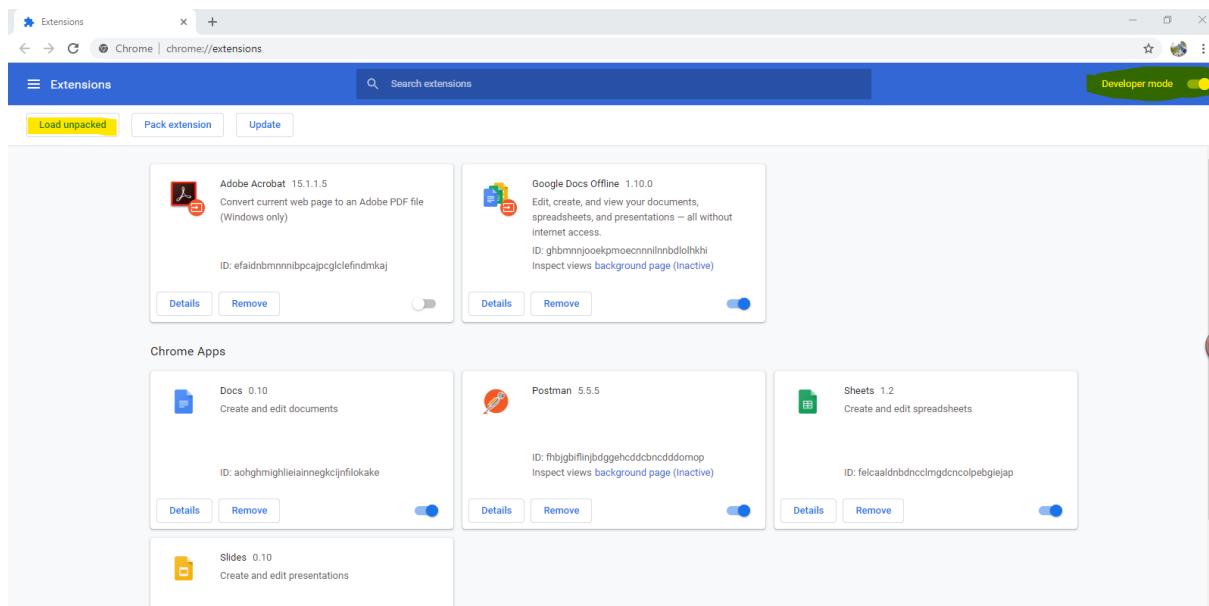
How to add 3DX-TAS Web Recorder extension on different browsers

- **Steps for Google Chrome:**

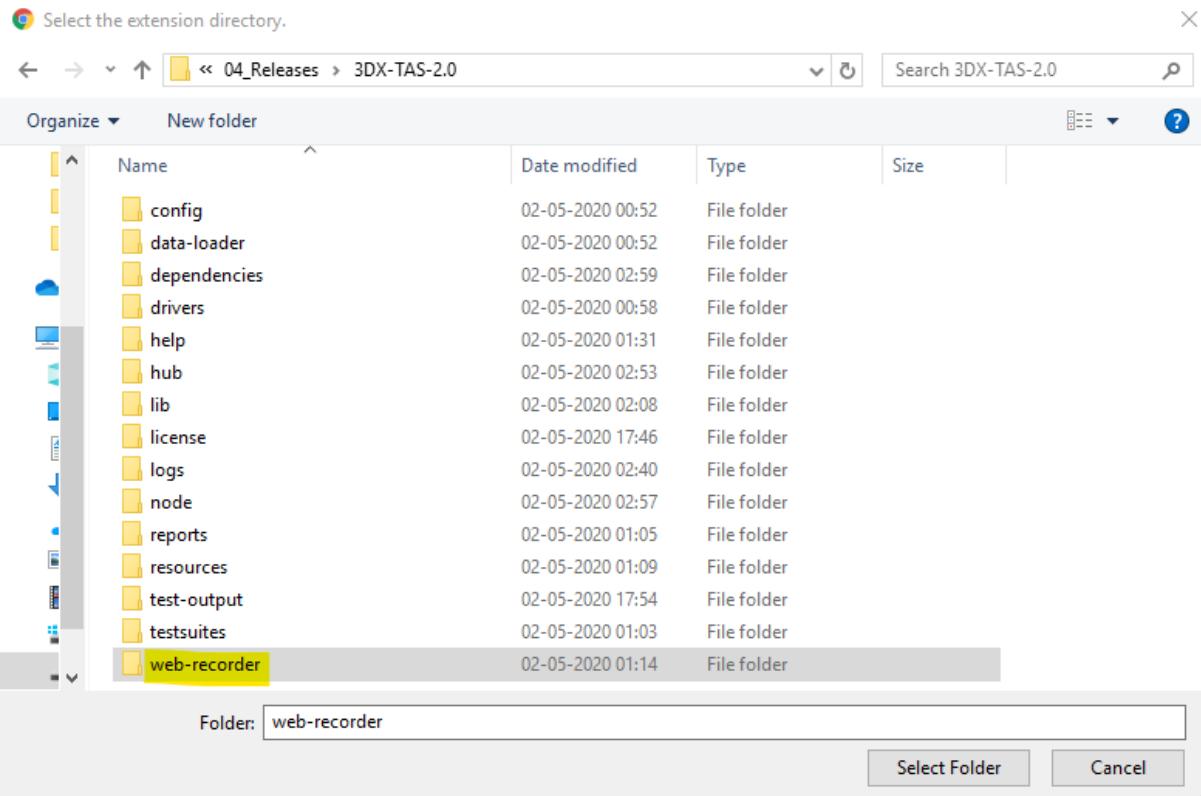
1. Open Extension tab on chrome browser
Go to More Tools and click on Extensions as shown in following screenshot



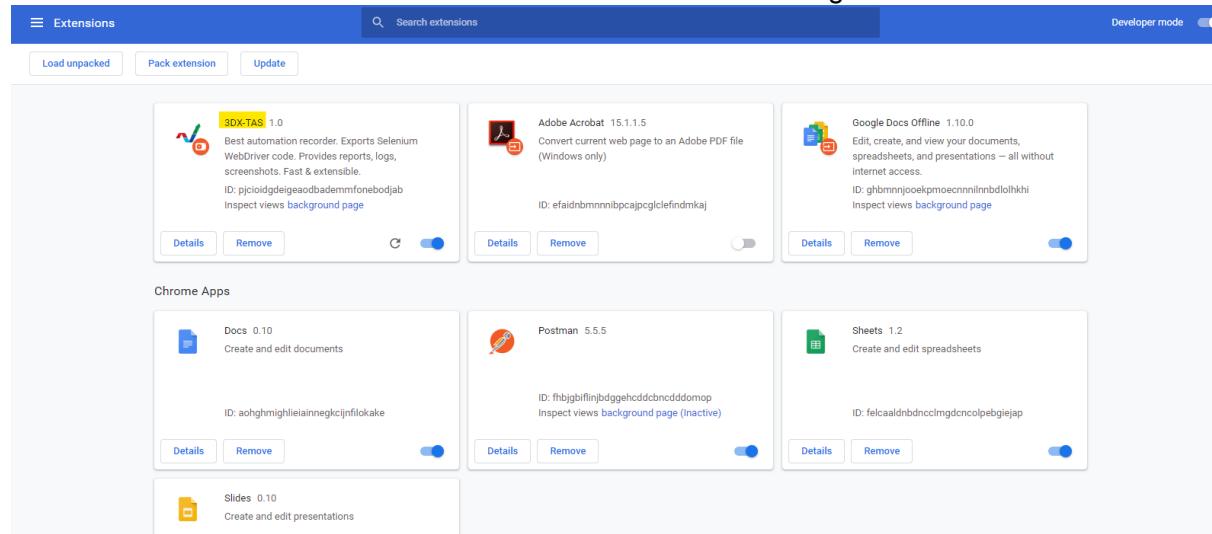
2. Enable Developer mode and Click on Load unpacked



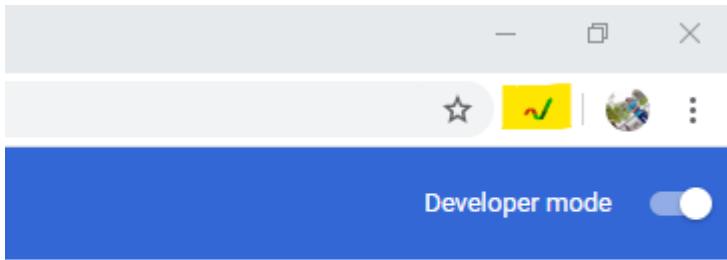
3. Select 3DX-TAS/web-recorder folder and click on Select Folder



After Select Folder extension will be added as shown in following screenshot



4. Disable Developer mode
5. Now 3DX-TAS Web Recorder is ready to use. An Icon will be shown on top of the browser. Clicking on that icon will open 3DX-TAS Web Recorder.



- **Steps for Mozilla Firefox:**

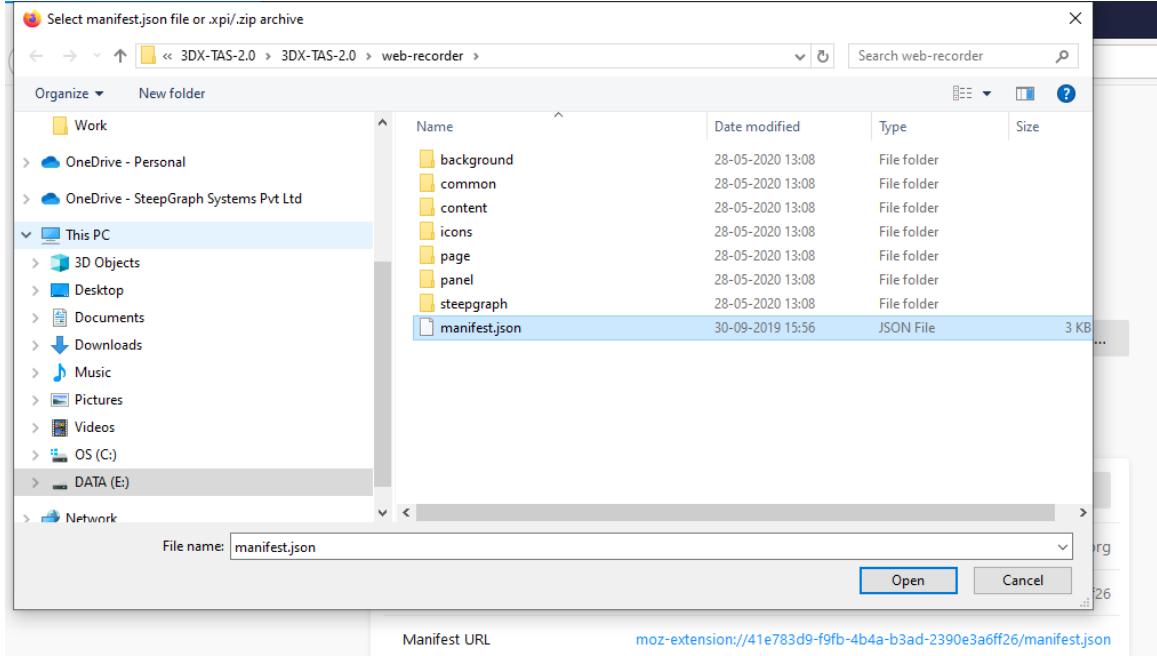
1. Open Firefox browser
2. From search bar, type "about:debugging" and search

The screenshot shows the 'about:debugging#setup' page in Mozilla Firefox. The URL in the address bar is `about:debugging#/setup`. The page has three main sections: 'Setup' (with a gear icon), 'Connect a Device' (with a USB icon), and 'Network Location' (with a network icon). In the 'Connect a Device' section, there is a note about enabling USB devices. On the left sidebar, there are links for 'Setup' and 'This Firefox'. The 'This Firefox' link is underlined and highlighted in blue, indicating it has been clicked.

3. Now from setup page find "This Firfeox" text in left and click

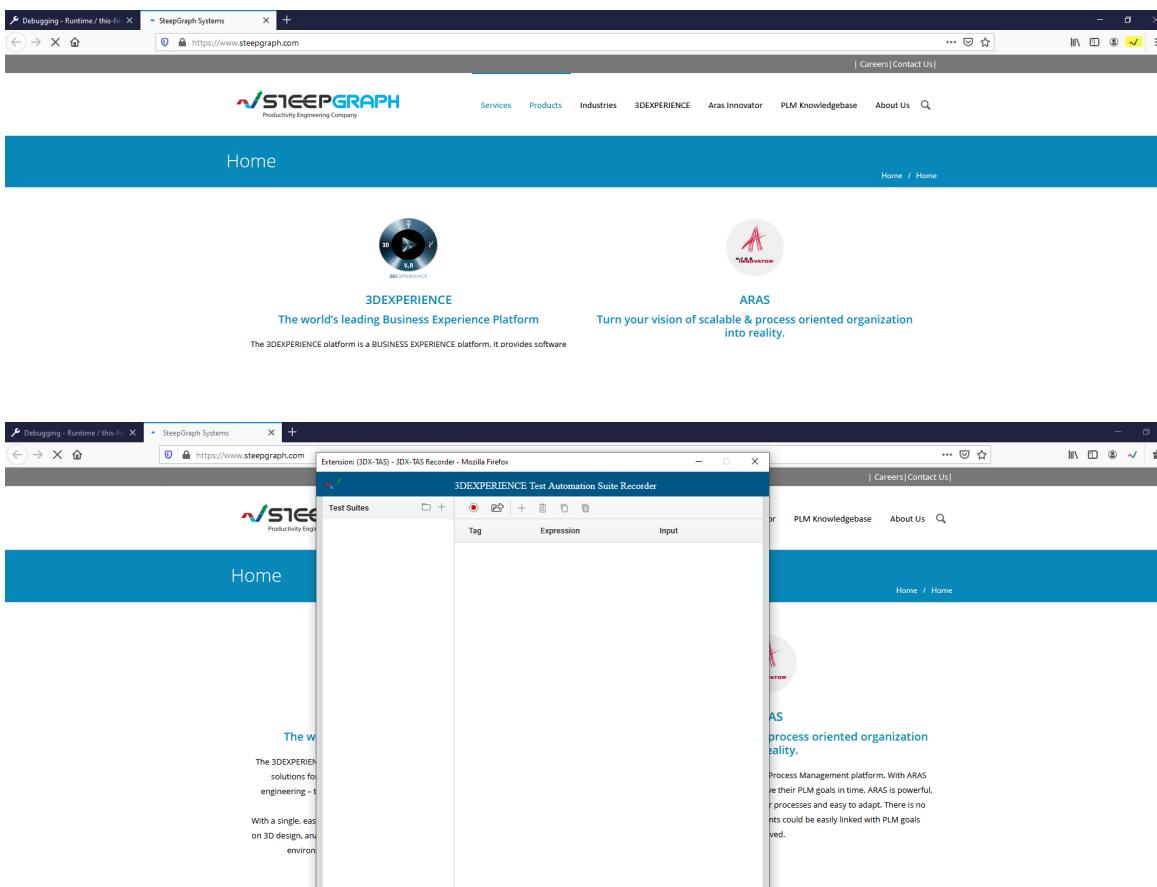
The screenshot shows the 'about:debugging#runtime/this-firefox' page in Mozilla Firefox. The URL in the address bar is `about:debugging#/runtime/this-firefox`. The page displays information about temporary and installed extensions. The 'Temporary Extensions (0)' section shows 'Nothing yet.' The 'Extensions (1)' section lists the 'Default' extension, which is identified by its Extension ID: `default-theme@mozilla.org`, Internal UUID: `41e783d9-f9fb-4b4a-b3ad-2390e3a6ff26`, and Manifest URL: `moz-extension://41e783d9-f9fb-4b4a-b3ad-2390e3a6ff26/manifest.json`. There is also an 'Inspect' button next to the extension entry.

4. Now click "Load Temporary Add-on" button, pop-up window is opened



5. Now find 'manifest_Firefox.json' file and click Open

6. 3DX_TAS extension is loaded and now we can see icon in right-upper toolbar

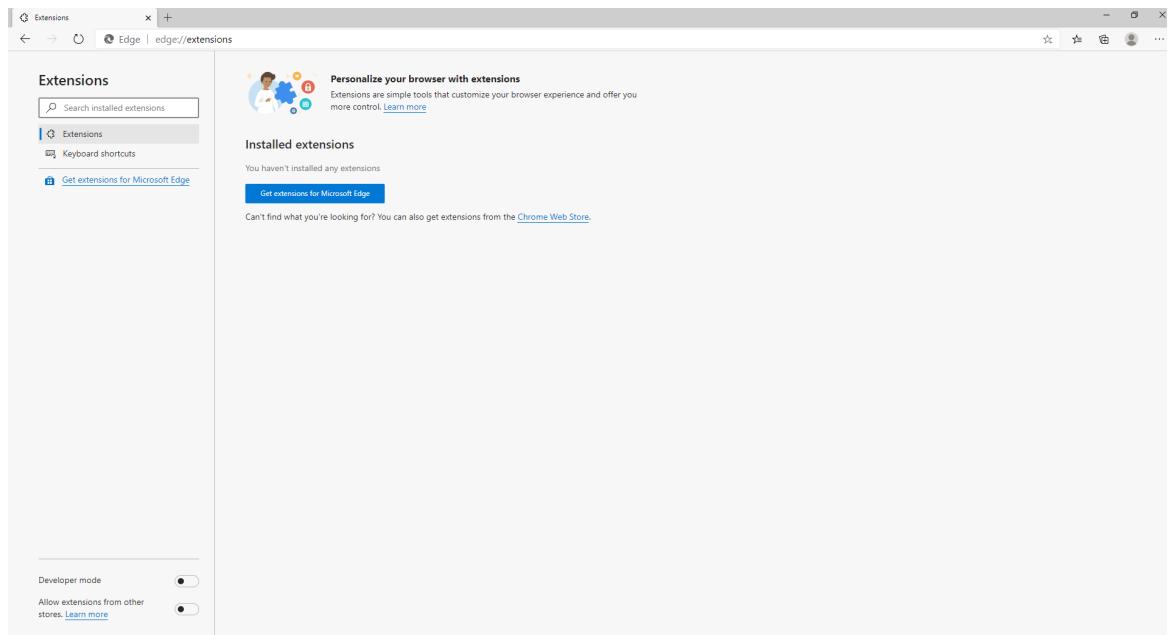


- **Steps for Microsoft Edge:**

1. Open Edge browser
2. From the upper right corner Click Setting and more icon(three dot in horizontal pattern)..



3. Navigate till Extensions and click.



4. Now from Extension page in Left-bottom corner ON the Developer mode

Extensions x +

← → ⏪ Edge | edge://extensions

Extensions

Search installed extensions

Extensions

Keyboard shortcuts

Get extensions for Microsoft Edge



Installed extensions

You haven't installed any extensions yet.

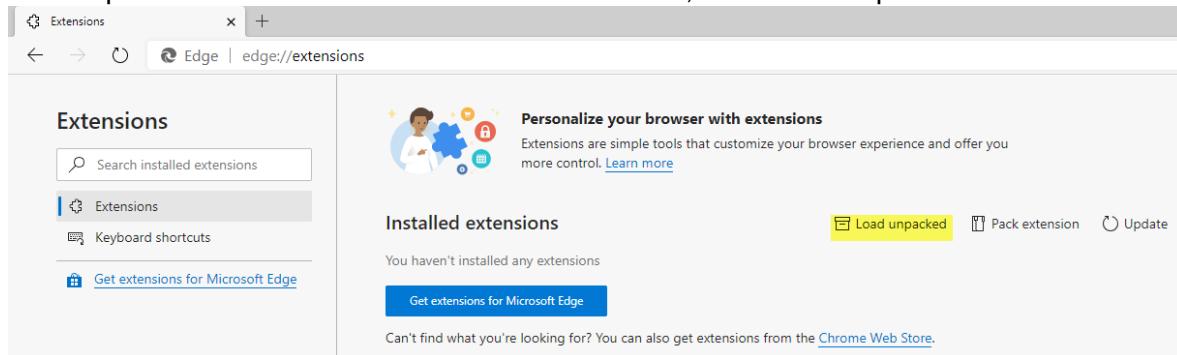
Get extensions

Can't find what you're looking for? Try searching.

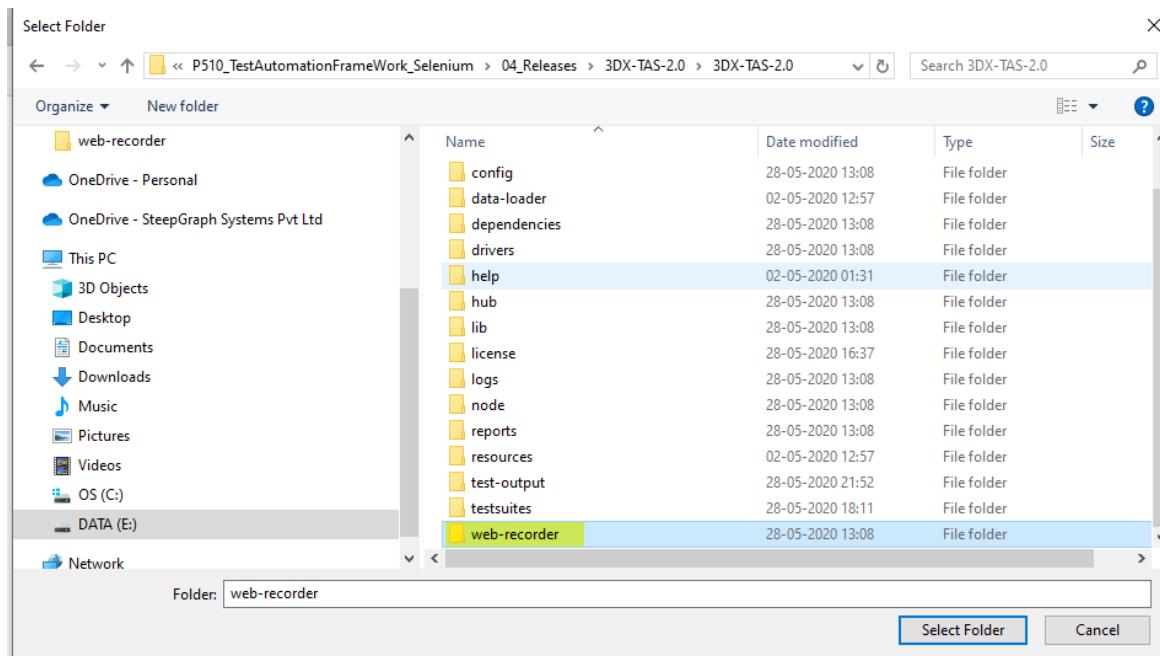
Developer mode

Allow extensions from other stores. [Learn more](#)

5. Three options are loaded next to Installed extensions, click 'Load unpacked' button

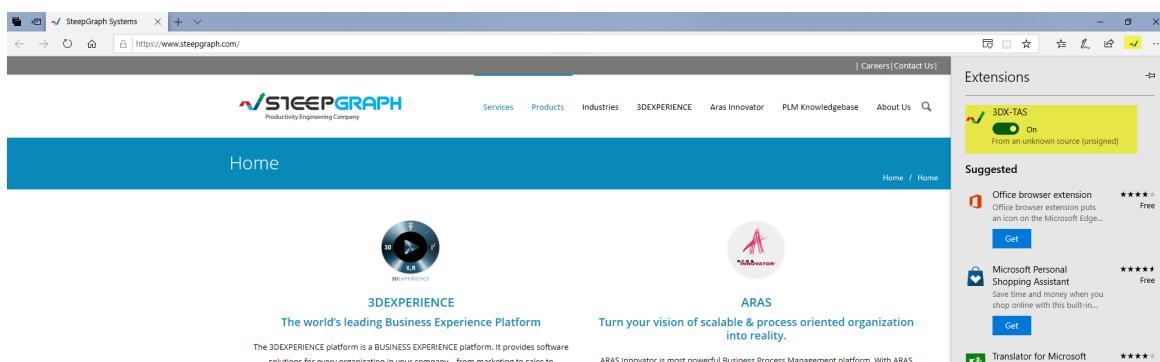


6. Pop-up window is opened, from that search the Recorder folder structure and click on Select Folder

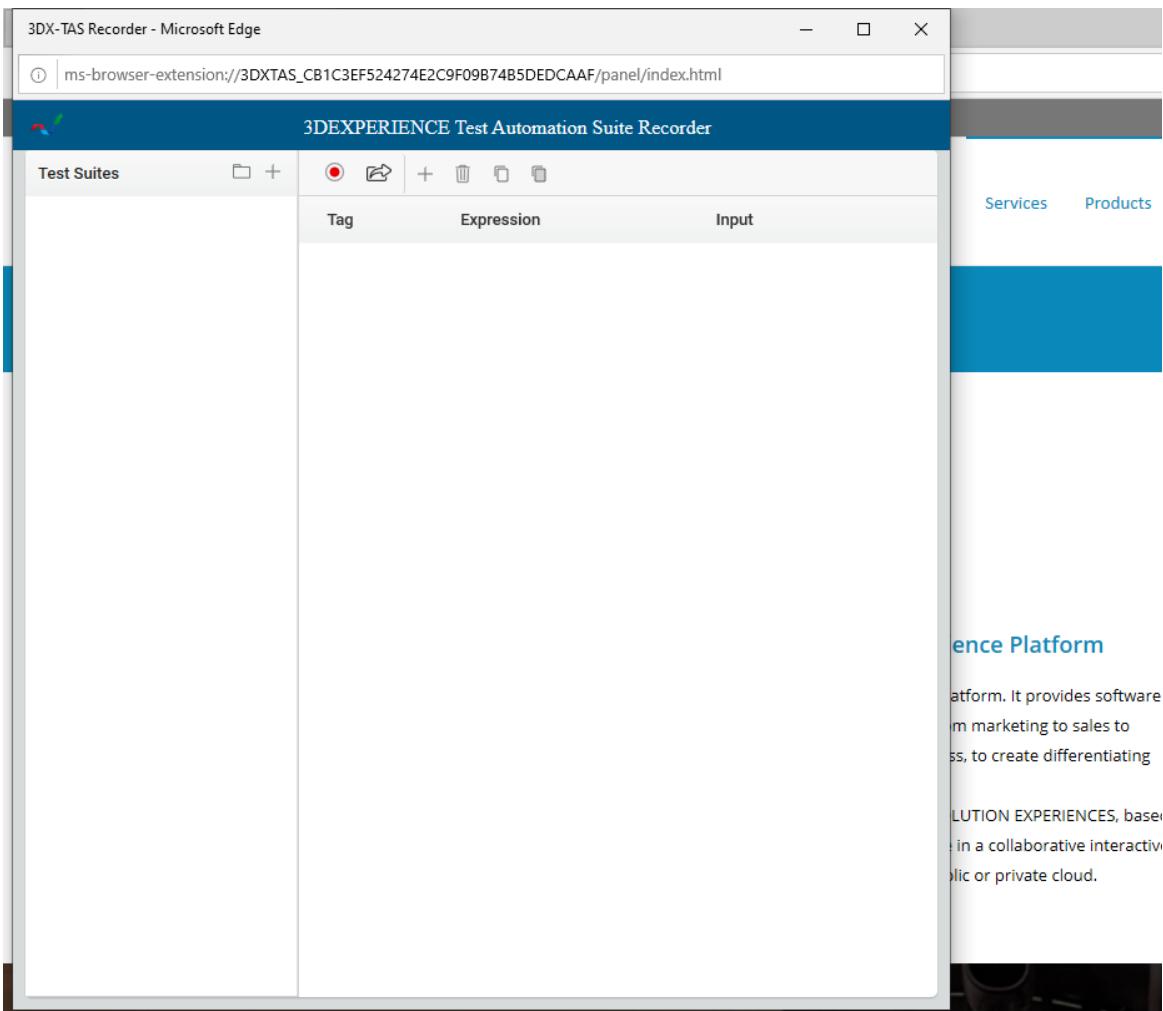


NOTE: Before selecting web-recorder folder, replace the content of manifest.json and panel\js\background\recorder.js with manifest_Edge.json and panel\js\background\recorder_Edge.js respectively. User could also refer to Readme file for help related to any of these issues.

7. 3DX_TAS extension is loaded and now we can see icon in right-upper toolbar



After opening recorder through Microsoft Edge:



How to use 3DX-TAS Web Recorder

1. Toolbar:

The toolbar consists of 8 buttons namely Open, New, Record, Play, Export, Delete, Copy and Paste. Each buttons perform different functionalities, which are provided below:

- **Open:**
Clicking on this button, will open the existing Test Case in the system.
- **New:**
Clicking on this button, will allow you to add a new test case inside the test suite.
- **Record:**
This button will let you record the test steps.
- **Export:**
Clicking on this button, would take you to the script mode, where you can choose the format. By default, 'XML' format is selected. Here you also have the option to save the script. Clicking on Save As File button, in the below screenshot will allow you to save the script as a Groovy file.
- **Add new test step:**
Clicking on 'Add new test step' button, will give you list of command you want to execute, as per business requirements.
- **Delete the current test steps:**
This button will delete the test case which is not required.
- **Copy the current test steps:**
This button will copy the selected step and store it to clipboard.

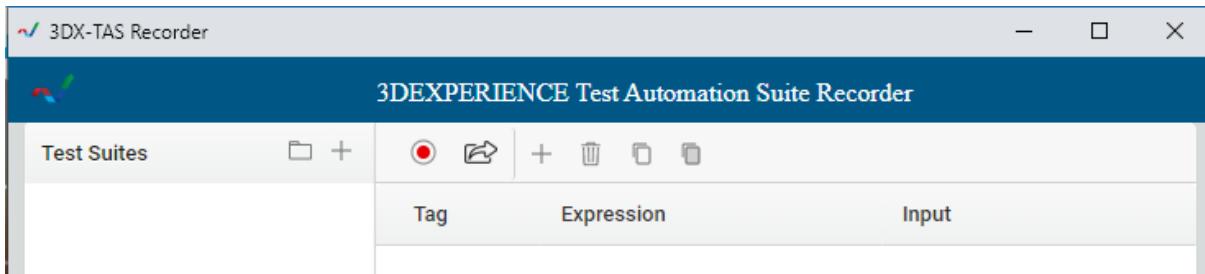
- **Paste the copied test steps as the next step of the current one:**
Select the step and click on the Paste button, copied step is pasted after the selected step.

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <!-- Script Generated In SteepGraph 3DX Plugin Format -->
3 <TestCase xmlns="https://www.steepgraph.com" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="https://www.steepgraph.com ../../resources/main/TestAutomationFramework.xsd">
4   <ClickElement locatorType="name" locatorExpression="username"/>
5   <InputText locatorType="name" locatorExpression="username" input="3dx_user1"/>
6   <ClickElement locatorType="name" locatorExpression="password"/>
7   <InputText locatorType="name" locatorExpression="password" input="Passport1"/>
8   <ClickElement locatorType="xpath" locatorExpression="//input[@value='Log in']"/>
9   <ClickElement locatorType="xpath" locatorExpression="//input[@type='text']"/>
10  <ClickElement locatorType="xpath" locatorExpression="link=Part_50927981_0001154_01"/>
11  <ClickElement locatorType="xpath" locatorExpression="//div[@id='tile']//div[2]//div[6]//div/div/div[5]"/>
12  <ClickElement locatorType="xpath" locatorExpression="//li[@id='action_DisplayDetails']/span[2]"/>
13 </TestCase>

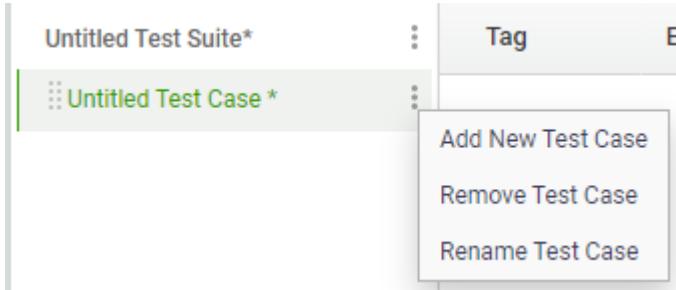
```

Copy to Clipboard Save As File... Close

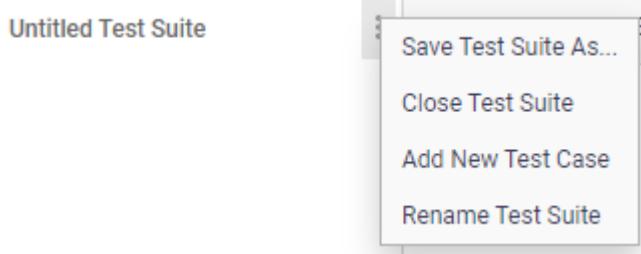


2. Test Case Explorer:

Test case explorer contains test suites. In a test suite, you can have multiple test cases. Right-clicking on Test case gives you three options – ‘Add New Test Case’, ‘Remove Test Case’, and ‘Rename Test Case’.



Right-clicking on Test Suite gives you four options – ‘Save Test Suite As’, ‘Close Test Suite’, ‘Add New Test Case’, and ‘Rename Test Suite’.



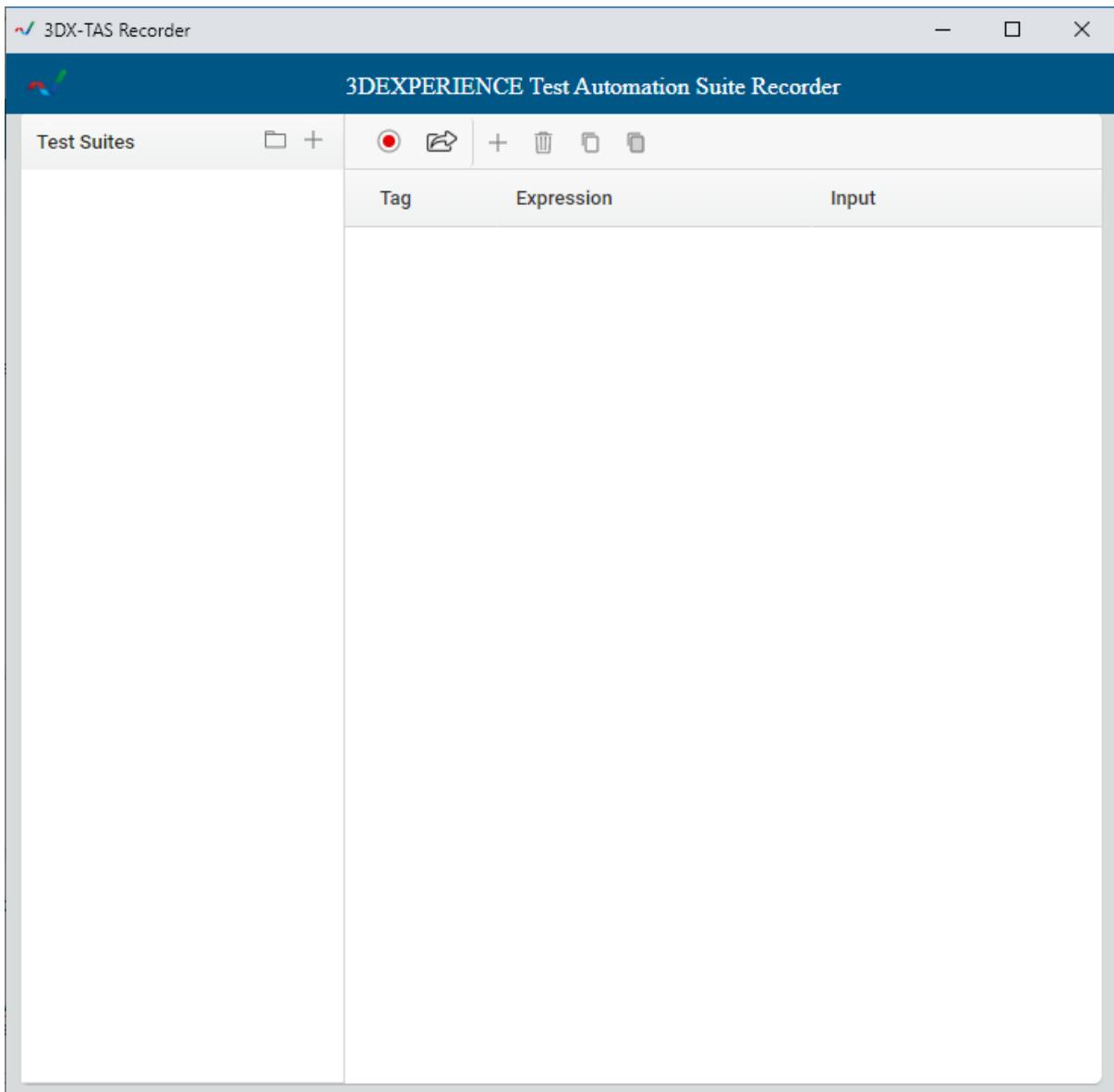
3. Test Case Window:

This area allows you to add, delete and edit test steps. Each test step contains Tag, Expression, and Input. You can also drag and drop to change the order of test steps.

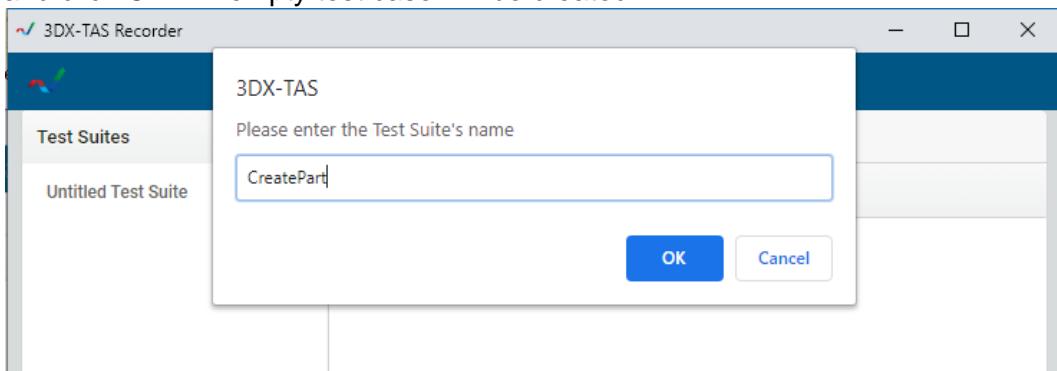
How to use Recorder:

1. Login to the given URL (<https://www.steepgraphdev.com/3dspace>) or any other URL, on which you want to automate the test scenario.

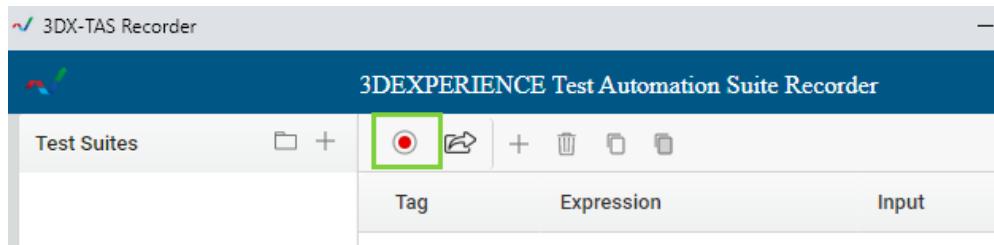
2. Now open the 3DX-TAS Recorder on chrome.



3. Click on the New Test Case button on the main toolbar. Provide a name for your test case and click OK. An empty test case will be created.



4. Click on the Record button to start recording test case.



5. Execute all your test steps on the browser. All of your actions performed on the browser will be recorded by 3DX-TAS Recorder. The recorded steps and related data are shown below, for the actions of going to steepgraphdev and searching Part on the browser.

6. You can Stop or Pause recording if needed.

7. Recorded objects and actions are saved into the test case window as shown below.

3DEXPERIENCE Test Automation Suite Recorder		
Test Suites	Tag	Expression
Untitled Test Suite*	open	https://www.steepgraphdev.co m/3dpassport/login?service=htt ps%3A%2Fwww.steepgraph dev.com%2F3dspace%2F
	click	name=username
	type	name=username 3dx_user1
	click	name=password
	type	name=password Passport1
	click	//input[@value='Log in']
	click	//input[@type='text']
	click	link=Part_50927981_0001154_0 1
	click	//div[@id='tile']/div[2]/div[6]/div/ div/div/div/div[5]
	click	//li[@id='action_DisplayDetails']/ span[2]

8. Now choose the test case you want to export as script, then select Export option.

9. After choosing the desired format, click on Save as file to export your test case.

Export Test Case as Script

Format **XML**

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <!-- Script Generated In SteepGraph 3DX Plugin Format -->
3 <TestCase xmlns="https://www.steepgraph.com" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="https://www.steepgraph.com ../../resources/main/TestAutomationFramework.xsd">
4   <ClickElement locatorType="name" locatorExpression="username"/>
5   <InputText locatorType="name" locatorExpression="username" input="3dx_user1"/>
6   <ClickElement locatorType="name" locatorExpression="password"/>
7   <InputText locatorType="name" locatorExpression="password" input="Passport1"/>
8   <ClickElement locatorType="xpath" locatorExpression="//input[@value='Log in']"/>
9   <ClickElement locatorType="xpath" locatorExpression="//input[@type='text']"/>
10  <ClickElement locatorType="xpath" locatorExpression="link=Part_50927981_0001154_01"/>
11  <ClickElement locatorType="xpath" locatorExpression="//div[id='tile']/div[2]/div[6]/div/div/div/div/div[5]"/>
12  <ClickElement locatorType="xpath" locatorExpression="//li[@id='action_DisplayDetails']/span[2]"/>
13 </TestCase>

```

Copy to Clipboard **Save As File...** **Close**

Created with the Personal Edition of HelpNDoc: [Free Web Help generator](#)

How To Encrypt Password

How to Encrypt Password using EncryptPassword.bat file

Following are the steps to Encrypt Password using notepad++:

1. Open EncryptPassword.bat file using notepad++

```

@echo off
java -cp lib/SGTAS-2.1.jar com.steepgraph.ta.framework.encryptUtil %1
pause

```

2. Replace %1 at the end of the line 2 with the username whose Password needs to be encrypted.

3. Save the file by pressing CTRL+s.

4. doubleClick on the batfile. then command prompt opens with encrypted password in it as shown below.

```

C:\WINDOWS\system32\cmd.exe
---Original Password---
manasak
---Encrypted Password---
eyLWNq11+chHPUDn/ox+ow==
Press any key to continue . . .

```

5. encrypted Password can be used as needed.

Created with the Personal Edition of HelpNDoc: [Produce online help for Qt applications](#)

Native App Recorder

The Test Automation Tool also includes 3DX Native App Recorder. With help of this Recorder, user could easily record different actions, by performing the same on their native/desktop application.

There are three phases of the recorder (This is general flow of 3DX Native App Recorder which is followed basically, once the 3DX Native App Recorder is launched successfully. Steps for launching 3DX Native App Recorder is provided after this topic) which are mentioned below:

- Start Recorder:** Once the 3DX Native App Recorder is successfully launched and the test suite folder is selected successfully, user needs to start the recorder by clicking on Record button, located below Toolbar Menu and at top of Tag Library. The image for the same is provided below:



- Record Scripts:** Once the recorder is started, user could perform different actions like Clicking different elements, input text using sendKeys etc.
- Stop Recording:** Once the user has completed recording of required actions, user could stop recording by using shortcut key i.e. shift+alt+c.
- Save Test Script:** Once the user has completed their script either by recording/manual editing, user could Save their test script by either clicking in **Save Test Script** button or either by using Shortcut Key(i.e. ctrl+s). The image for same is provided below:



- **Guidelines:**

- User could use this recorder, in order to develop the basic script for their business requirement. It makes the script accurately in most the cases but in some cases, user might need to edit script as it may consist of elements/objects which are dynamic in nature resulting in different element/object name and properties from time to time. In such cases user could use RegisterObject or FindElement Tags etc. in order to capture dynamic elements and their varying properties.
- In case of dynamic elements, user might need to add data or edit data in the csv .The screenshot for the same is provided below:

filepath	username	password	checkbox1	PhysicalPr	Prod	Text
C:\\\\Programs\\3dx_user2	Passport1	Rememe	PhysicalPr	\$reg{Phys	GLOBAL	

- User might need to perform different mouse operations(such as moving cursor from one position to another, clicking Element etc.), input operations(such as sending keys) at normal speed, instead of performing the same at fast speed, which might result in discrepancies while recording.
- While creating Building Block, the images folder should be created under Building Blocks folder for placing the images captured for scripts. This is applicable for general Tags. For 3DX tags, the images should be replaced under common images folder.

- **How to start 3DX Native App Recorder:**

- SG-TAS Package**

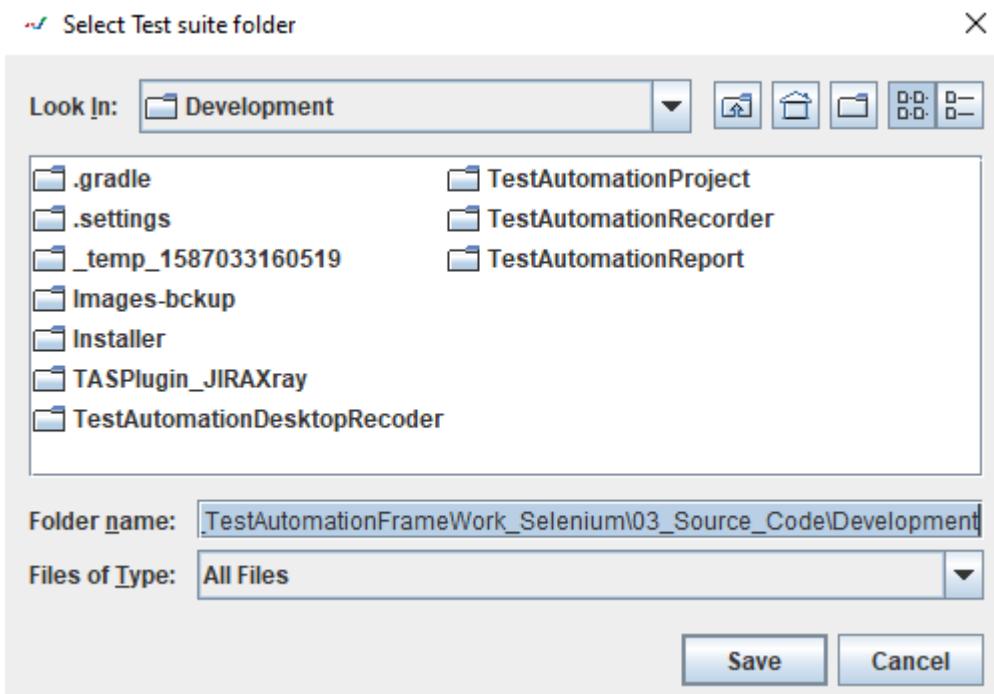
Following is the installed folder structure of SG-TAS-2.1:

 config	14-10-2024 17:45	File folder
 data-loader	22-05-2024 16:41	File folder
 dependencies	14-10-2024 17:59	File folder
 documents	08-10-2024 03:42	File folder
 drivers	14-10-2024 17:40	File folder
 hub	24-09-2024 00:54	File folder
 lib	14-10-2024 17:59	File folder
 license	13-09-2024 00:50	File folder
 logs	14-10-2024 17:16	File folder
 node	24-09-2024 00:54	File folder
 reports	22-05-2024 16:41	File folder
 resources	24-09-2024 00:54	File folder
 test-output	14-10-2024 17:16	File folder
 testsuites	14-10-2024 17:16	File folder
 web-recorder	22-05-2024 16:42	File folder
 EncryptPassword.bat	22-05-2024 16:41	Windows Batch File
 EncryptPassword.sh	24-09-2024 00:54	SH Source File
 End_SG-TAS_Execution.bat	13-09-2024 00:50	Windows Batch File
 Execute_SG_TAS.bat	14-10-2024 17:40	Windows Batch File
 Execute_SG_TAS.sh	24-09-2024 00:54	SH Source File
 Execute_SG_TAS_NativeRecorder.bat	24-09-2024 00:54	Windows Batch File
 Execute_SG_TAS_NativeRecorder.sh	24-09-2024 00:54	SH Source File
 LogFileCompressor_Windows.bat	13-09-2024 00:52	Windows Batch File
 Stop_SG_TAS.bat	22-05-2024 16:41	Windows Batch File

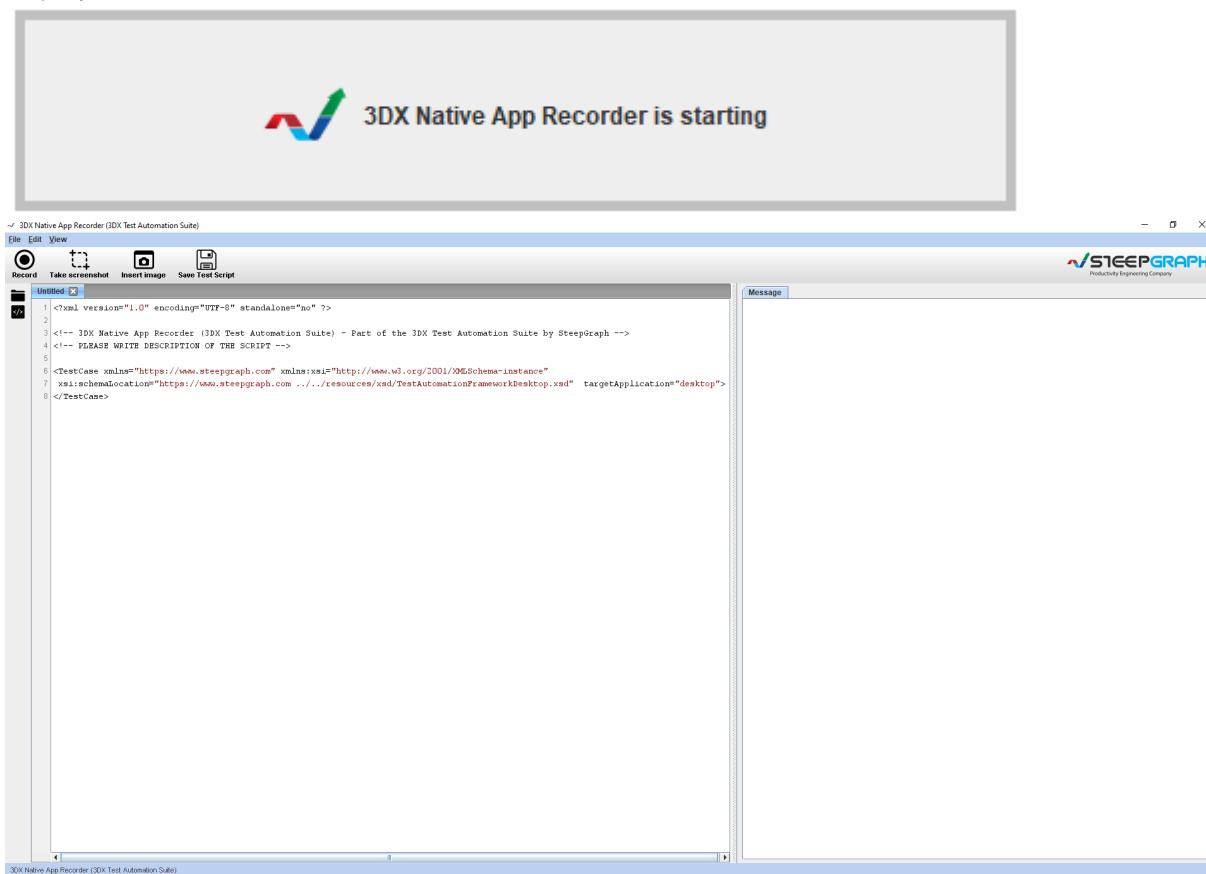
2. After successful extraction, user needs to generate license from license folder available in the extracted folder..
3. Once user has obtained the license, user need to copy/paste **license.xml** in license folder and should right click on **Execute_SG_TAS_NativeRecorder** and open **Terminal**, by clicking on **Terminal option**, from "Show in Local Terminal" menu. Once user is navigated to **Terminal**, user needs to enter command "**Execute_SG_TAS_NativeRecorder.bat**" in Terminal, in order to start the Recorder.
4. User needs to **Select Test Suite folder** from the list of folders available, in order to store all the test cases in selected test-suite folder and click on **Save** button. The screenshot for the same is provided below:

NOTE:

- o It is recommended to select "testsuites" folder from the list of folders available, as user need the recorded scripts in the same folder later on for script execution. The same path needs to be mentioned in **TestPlan.xml** later.
- o If user clicks on Cancel button or "X" symbol, the recorder will not be opened and recorder instance would get stopped.



5. After selecting the Test Suite folder, user would be displayed message stating "**SG Native App Recorder is starting**" and recorder will be opened displaying different **ui components**(i.e. **Toolbar Menu, Tag Library, XML Script Editor, buttons such as Record, Take screenshot and Insert Image and Message Section**) The screenshot for the same is displayed below:



- **UI Components:**

1. Toolbar Menu:

The toolbar consists of 3 menu's i.e. File, Edit and View, which further have consists of different options, with help of which user could perform different functions related to their script manipulation(which could relate to creating new script, editing current script or modifying scripts). The toolbar menu's along with their options are explained below:

- a) **File:** This menu contains operations/options related to creating a new XML script, opening existing XML script and closing current XML script. The options which is present under File menu is as follows:
 - i) **New:** When user wants/chooses to create new script, user needs to click on **New** option or by using shortcut key(**i.e. Ctrl+N**). The new script always gets created with "**Untitled**" name. The screenshot for the same is provided below:

```

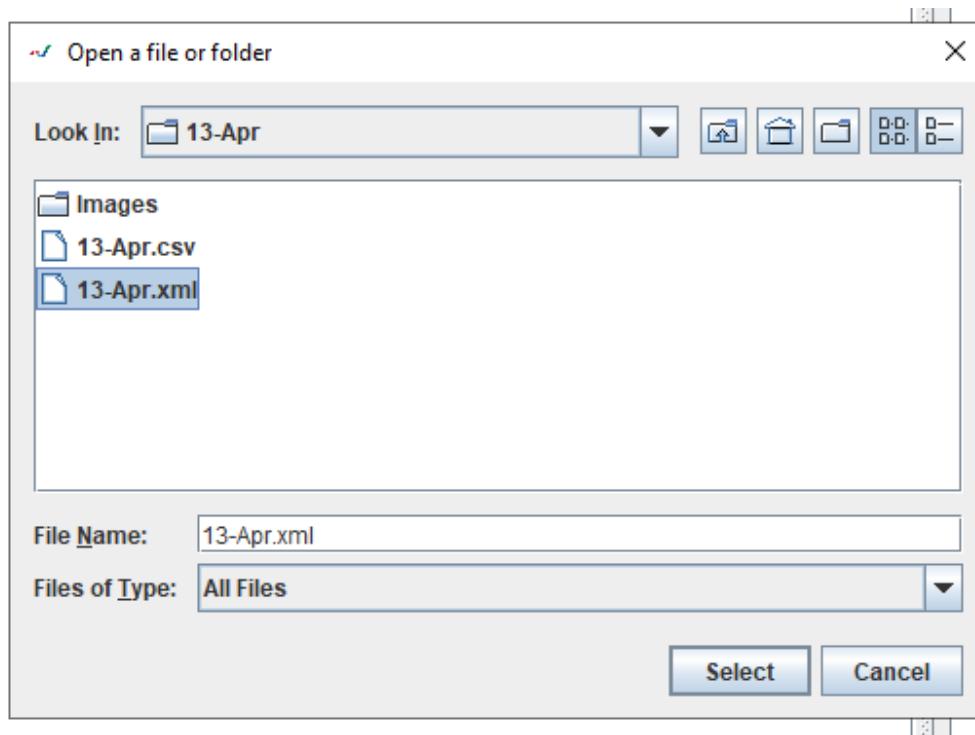
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<!-- 3DX Native App Recorder (3DX Test Automation Suite) - Part of the 3DX Test Automation Suite by SteepGraph -->
<!-- PLEASE WRITE DESCRIPTION OF THE SCRIPT -->
<TestCase xmlns="https://www.steepgraph.com" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="https://www.steepgraph.com ../../resources/xsd/TestAutomationFrameworkDesktop.xsd" targetApplication="desktop">
</TestCase>

```

- ii) **Open:** When user want to open already existed/created XML script, user needs to click on **Open** option or by using shortcut key(**i.e.Ctrl+O**). Upon clicking **Open** option, user would be displayed "**Open a file or folder**" dialogue box requesting user to select the XML script which user wants to open with "**Select**" and "**Cancel**" as the options. The screenshot for the same is provided below:

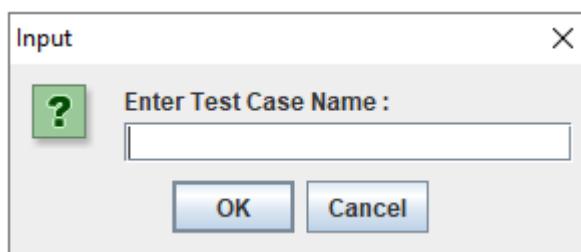
NOTE:

- 1) User needs to select XML script file(i.e. .xml extension file) in order to open already existing xml script.
- 2) User could either open XML script by either double clicking on .xml file or by selecting the same and clicking on Select Option. If user chooses to cancel file opening operation, User could do same by clicking Cancel button.

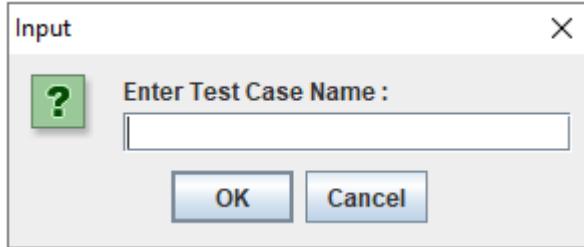


- iii) **Save:** When user want to save the script, user needs to either click on **Save** option or by using shortcut key(**Ctrl+S**). Upon clicking Save button, user would be popped up with "Input" dialogue box requesting user to enter Test Case Name with "**OK**" and "**Cancel**" as the options. User need to enter the Test Case name and click on **OK** button in order to save the script. If user wishes to cancel the Save operation, user could click on **Cancel** option. The screenshot for the same is provided below:

NOTE: If user chooses to Click on OK button without entering Test Case Name, the Save operation will not work and user would need to save the script again.



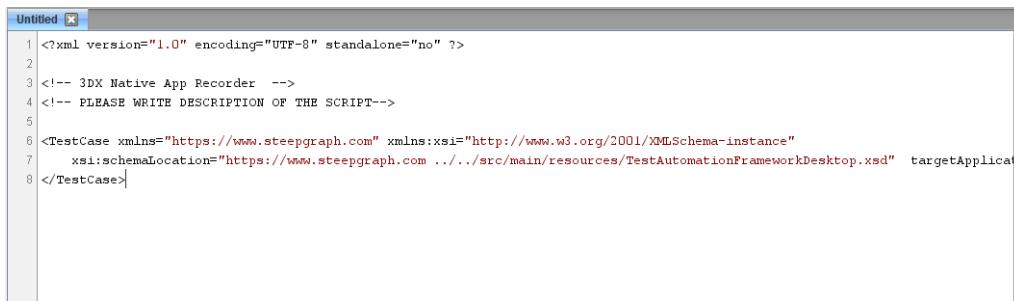
- iv) **Save as... :** When user want to save already existing script or edited script with new name, user need to either click on **Save as...** option or by using shortcut key(i.e. **Ctrl+Shift+S**). user would be popped up with "Input" dialogue box requesting user to enter Test Case Name with "**OK**" and "**Cancel**" as the options. User need to enter the Test Case name and click on **OK** button in order to save the script. If user wishes to cancel the Save operation, user could click on **Cancel** option. The screenshot for the same is provided below:
NOTE: If user chooses to Click on OK button without entering Test Case Name, the Save operation will not work and user would need to save the script again.



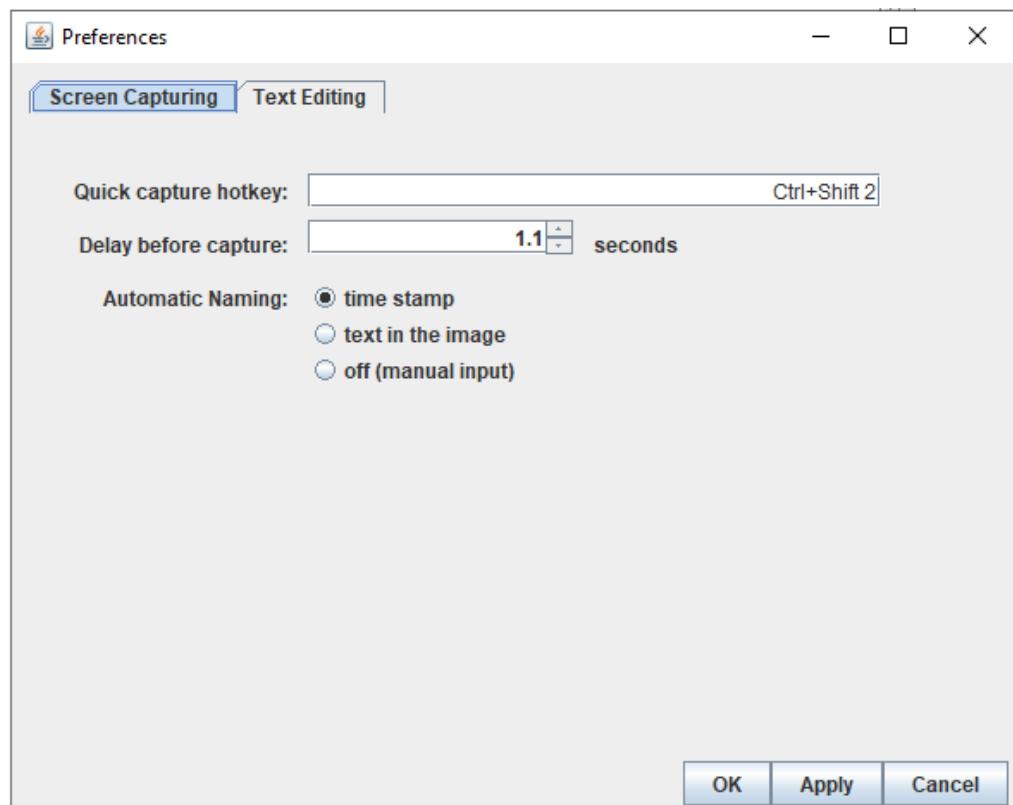
- v) **Close Tab:** When user chooses to close current Tab of XML Script Editor(i.e. where user is currently working on), then user need to either click on **Close Tab** option or by using shortcut key(i.e. **Ctrl+W**). The screenshot for the same is provided below:



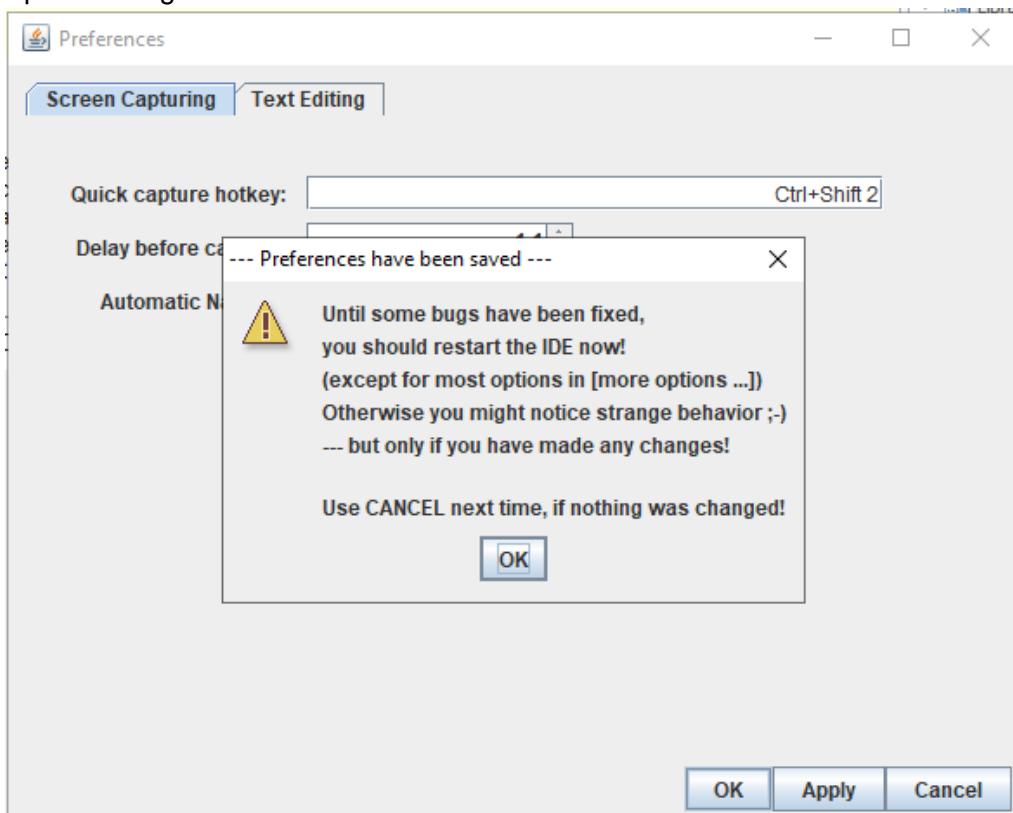
When user clicks on **Close Tab** option:



- vi) **Preferences:** When user wants to edit 3DX Native App Recorder settings related to Screen Capturing(such as delay time before capturing or Quick capture shortcut key etc.) and Text Editing(such as Font name, Font size or Indentation setting), user need to either click on **Preferences** option or using shortcut key(i.e.**Ctrl+P**) with "OK", "Apply" and "Cancel" as the options. User could modify required settings and click on **Apply** button and then click on **OK** button in order save modified settings. User would be displayed "**--- Preferences have been saved ---**" pop-up displaying message to restart the recorder, so that new setting could be applied to Recorder as per user modification with **OK** button as option. User needs to click on **OK** button for confirmation. User could click on **Cancel** button in case he/she wishes to cancel changes user made in Preferences. The screenshot for the same is provided below:



Upon clicking **OK** button:



- vii) **Quit:** When user wants to quit the 3DX Native App Recorder, user could click on **Quit** button.
- b) **Edit:** This menu contains operations/options related to XML script content modification/searching. The options which is present under **Edit** menu is as follows:

- i) **Undo:** When user have mistakenly entered wrong value or text manually and user wants to revert his/her changes, then user need to click on **Undo** button or by using shortcut key(i.e.Ctrl+Z). Upon clicking **Undo** button, the changes would be reverted. The screenshot for the same is provided below:



```

1 <?xml version="1.0" encoding="UTF-8" standalone="no" ?>
2
3 <!-- 3DX Native App Recorder -->
4 <!-- PLEASE WRITE DESCRIPTION OF THE SCRIPT-->
5
6 <TestCase xmlns="https://www.steepgraph.com" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
7   xsi:schemaLocation="https://www.steepgraph.com ../../src/main/resources/TestAutomationFrameworkDesktop.xsd" targetApplication="3DX Native App Recorder">
8
9   <!-- TBD : Step description -->
10  <Assert criteria="" locatorImage="" timeout="" text="" colorDifference="" l="" r="" t="" b="" />
11  Test
12
13
14 </TestCase>

```

After clicking **Undo** option:



```

1 <?xml version="1.0" encoding="UTF-8" standalone="no" ?>
2
3 <!-- 3DX Native App Recorder -->
4 <!-- PLEASE WRITE DESCRIPTION OF THE SCRIPT-->
5
6 <TestCase xmlns="https://www.steepgraph.com" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
7   xsi:schemaLocation="https://www.steepgraph.com ../../src/main/resources/TestAutomationFrameworkDesktop.xsd" targetApplication="3DX Native App Recorder">
8
9   <!-- TBD : Step description -->
10  <Assert criteria="" locatorImage="" timeout="" text="" colorDifference="" l="" r="" t="" b="" />
11
12
13
14 </TestCase>

```

- ii) **Redo:** When user have mistakenly applied **Undo** operation on script and user wants to revert his/her changes happened due to **Undo** operation , then user need to click on **Redo** button or by using shortcut key(i.e.Ctrl+Shift+Z). Upon clicking **Redo** button, the changes happened due to **Undo** would be reverted. The screenshot for the same is provided below:



```

1 <?xml version="1.0" encoding="UTF-8" standalone="no" ?>
2
3 <!-- 3DX Native App Recorder -->
4 <!-- PLEASE WRITE DESCRIPTION OF THE SCRIPT-->
5
6 <TestCase xmlns="https://www.steepgraph.com" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
7   xsi:schemaLocation="https://www.steepgraph.com ../../src/main/resources/TestAutomationFrameworkDesktop.xsd" targetApplication="3DX Native App Recorder">
8
9   <!-- TBD : Step description -->
10  <Assert criteria="" locatorImage="" timeout="" text="" colorDifference="" l="" r="" t="" b="" />
11  Test
12 </TestCase>

```

When user by mistakenly clicks on **Undo** button:



```

1 <?xml version="1.0" encoding="UTF-8" standalone="no" ?>
2
3 <!-- 3DX Native App Recorder -->
4 <!-- PLEASE WRITE DESCRIPTION OF THE SCRIPT-->
5
6 <TestCase xmlns="https://www.steepgraph.com" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
7   xsi:schemaLocation="https://www.steepgraph.com ../../src/main/resources/TestAutomationFrameworkDesktop.xsd" targetApplication="3DX Native App Recorder">
8
9   <!-- TBD : Step description -->
10  <Assert criteria="" locatorImage="" timeout="" text="" colorDifference="" l="" r="" t="" b="" />
11
12 </TestCase>

```

When user clicks on **Redo** button:



```

1 <?xml version="1.0" encoding="UTF-8" standalone="no" ?>
2
3 <!-- 3DX Native App Recorder -->
4 <!-- PLEASE WRITE DESCRIPTION OF THE SCRIPT-->
5
6 <TestCase xmlns="https://www.steepgraph.com" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
7   xsi:schemaLocation="https://www.steepgraph.com ../../src/main/resources/TestAutomationFrameworkDesktop.xsd" targetApplication="3DX Native App Recorder">
8
9   <!-- TBD : Step description -->
10  <Assert criteria="" locatorImage="" timeout="" text="" colorDifference="" l="" r="" t="" b="" />
11  Test
12 </TestCase>

```

- iii) **Copy:** When user want to copy particular text/line, user need to select text/line by either clicking on **Copy** option or by clicking shortcut key(**i.e.Ctrl+C**), the selected text/line will get copied on clipboard, in order to paste the same in script at some other position. The screenshot for the same is provided below:



```

1 <?xml version="1.0" encoding="UTF-8" standalone="no" ?>
2
3 <!-- 3DX Native App Recorder -->
4 <!-- PLEASE WRITE DESCRIPTION OF THE SCRIPT-->
5
6 <TestCase xmlns="https://www.steepgraph.com" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
7   xsi:schemaLocation="https://www.steepgraph.com ../../src/main/resources/TestAutomationFrameworkDesktop.xsd" targetApplication="3DX Native App Recorder">
8
9   <!-- TBD : Step description -->
10  <Assert criteria="" locatorImage="" timeout="" text="" colorDifference="" l="" r="" t="" b="" />
11
12
13  <!-- TBD : Step description -->
14  <FileChooser locatorImage="" filenames="" filepath="" />
15 </TestCase>

```

- iv) **Copy line:** When user want to copy particular line, user needs to move cursor on that line by either click on **Copy line** option or by clicking shortcut key(**i.e.Ctrl+Shift+C**), the line will get selected and copied on clipboard, in order to paste the same in script at some other position. The screenshot for the same is provided below:



```

1 <?xml version="1.0" encoding="UTF-8" standalone="no" ?>
2
3 <!-- 3DX Native App Recorder -->
4 <!-- PLEASE WRITE DESCRIPTION OF THE SCRIPT-->
5
6 <TestCase xmlns="https://www.steepgraph.com" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
7   xsi:schemaLocation="https://www.steepgraph.com ../../src/main/resources/TestAutomationFrameworkDesktop.xsd" targetApplication="3DX Native App Recorder">
8
9   <!-- TBD : Step description -->
10  <Assert criteria="" locatorImage="" timeout="" text="" colorDifference="" l="" r="" t="" b="" />
11  <!-- TBD : Step description -->
12  <FileChooser locatorImage="" filenames="" filepath="" />
13 </TestCase>

```

- v) **Cut:** When user want to cut particular text/line, user need to select text/line by either clicking on **Cut** option or by clicking shortcut key(**i.e.Ctrl+X**), the selected text/line will get cut from the script, in order to paste the same in script at some other position. The screenshot for the same is provided below:



```

1 <?xml version="1.0" encoding="UTF-8" standalone="no" ?>
2
3 <!-- 3DX Native App Recorder -->
4 <!-- PLEASE WRITE DESCRIPTION OF THE SCRIPT-->
5
6 <TestCase xmlns="https://www.steepgraph.com" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
7   xsi:schemaLocation="https://www.steepgraph.com ../../src/main/resources/TestAutomationFrameworkDesktop.xsd" targetApplication="3DX Native App Recorder">
8
9   <!-- TBD : Step description -->
10  <Assert criteria="" locatorImage="" timeout="" text="" colorDifference="" l="" r="" t="" b="" />
11  <!-- TBD : Step description -->
12  <FileChooser locatorImage="" filenames="" filepath="" />
13 </TestCase>

```

When user cut the selected line:



```

1 <?xml version="1.0" encoding="UTF-8" standalone="no" ?>
2
3 <!-- 3DX Native App Recorder -->
4 <!-- PLEASE WRITE DESCRIPTION OF THE SCRIPT-->
5
6 <TestCase xmlns="https://www.steepgraph.com" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
7   xsi:schemaLocation="https://www.steepgraph.com ../../src/main/resources/TestAutomationFrameworkDesktop.xsd" targetApplication="Test Automation Framework Desktop">
8
9   <!-- TBD : Step description -->
10  <Assert criteria="" locatorImage="" timeout="" text="" colorDifference="" l="" r="" t="" b="" />
11  <!-- TBD : Step description -->
12 </TestCase>

```

- vi) **Cut line:** When user want to cut particular line, user needs to move cursor on that line by either click on **Cut line** option or by clicking shortcut key(**i.e.Ctrl+Shift+X**), the line will get selected and will be cut from the script, in order to paste the same in script at some other position. The screenshot for the same is provided below:



```

1 <?xml version="1.0" encoding="UTF-8" standalone="no" ?>
2
3 <!-- 3DX Native App Recorder -->
4 <!-- PLEASE WRITE DESCRIPTION OF THE SCRIPT-->
5
6 <TestCase xmlns="https://www.steepgraph.com" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
7   xsi:schemaLocation="https://www.steepgraph.com ../../src/main/resources/TestAutomationFrameworkDesktop.xsd" targetApplication="Test Automation Framework Desktop">
8
9   <!-- TBD : Step description -->
10  <Assert criteria="" locatorImage="" timeout="" text="" colorDifference="" l="" r="" t="" b="" />
11  <!-- TBD : Step description -->
12 </TestCase>

```

When user selected **Cut** option:



```

1 <?xml version="1.0" encoding="UTF-8" standalone="no" ?>
2
3 <!-- 3DX Native App Recorder -->
4 <!-- PLEASE WRITE DESCRIPTION OF THE SCRIPT-->
5
6 <TestCase xmlns="https://www.steepgraph.com" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
7   xsi:schemaLocation="https://www.steepgraph.com ../../src/main/resources/TestAutomationFrameworkDesktop.xsd" targetApplication="Test Automation Framework Desktop">
8
9   <!-- TBD : Step description -->
10  <!-- TBD : Step description -->
11 </TestCase>

```

- vii) **Paste:** When user wants to paste the already copied/cut text/line, user need to move cursor to line where he/she wants to paste the same by either clicking on **Paste** option or by clicking shortcut key(**i.e.Ctrl+V**), the copied/cut text/line would get pasted. The screenshot for the same is provided below:



```

1 <?xml version="1.0" encoding="UTF-8" standalone="no" ?>
2
3 <!-- 3DX Native App Recorder -->
4 <!-- PLEASE WRITE DESCRIPTION OF THE SCRIPT-->
5
6 <TestCase xmlns="https://www.steepgraph.com" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
7   xsi:schemaLocation="https://www.steepgraph.com ../../src/main/resources/TestAutomationFrameworkDesktop.xsd" targetApplication="Test Automation Framework Desktop">
8   <!-- TBD : Step description -->
9   <Action refId="" name="" locatorImage="" value="" />
10 </TestCase>

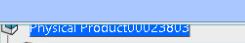
```

When user clicks on **Paste** option:

```
*Untitled X
1 <?xml version="1.0" encoding="UTF-8" standalone="no" ?>
2
3 <!-- 3DX Native App Recorder -->
4 <!-- PLEASE WRITE DESCRIPTION OF THE SCRIPT-->
5
6 <TestCase xmlns="https://www.steepgraph.com" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
7   xsi:schemaLocation="https://www.steepgraph.com ../../src/main/resources/TestAutomationFrameworkDesktop.xsd" targetApplication="Test Automation Framework Desktop">
8   <!-- TBD : Step description -->
9   <Action refId="" name="" locatorImage="" value="" />
10
11 <!-- TBD : Step description -->
12   <Action refId="" name="" locatorImage="" value="" />
13 </TestCase>
```

- viii) **Select All:** When user wants to select all the content of the script, user need to either click on **Select All** or by using shortcut key(i.e. **Ctrl+A**), all the content will get selected. The screenshot for the same is provided below:

81 <!-- TBD : Step description -->
82 <Wait time="2000" />
83
84 <!-- TBD : Step description -->
85 <ClickElement locatorImage="PM_TextDemo3_Subtype" " />

86
87 <!-- TBD : Step description -->
88 <Wait time="2000" />
89
90 <!-- TBD : Step description -->
91 <Action name="rightclick" locatorImage="Physical Product00023804" " />

92
93 <!-- TBD : Step description -->
94 <Wait time="2000" />
95
96 <!-- TBD : Step description -->
97 <ClickElement locatorImage="TICKET" " />

98
99 <!-- TBD : Step description -->
100 <Wait time="2000" />
101
102
103 <!-- TBD : Step description -->
104 <ClickElement locatorImage="PHYSICAL PRODUCT00023804" " />

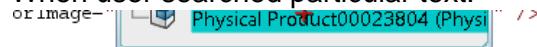
105
106
107 <!-- TBD : Step description -->
108 <SendKeys key="TD" />
109
110 <!-- TBD : Step description -->
111 <Wait time="2000" />
112
113 <!-- TBD : Step description -->
114 <ClickElement locatorImage="View" " />

115 </TestCase>

- ix) **Find:** When user wants to find some text within the script, user need to either click on **Find** option or by using shortcut key(**i.e.Ctrl+F**), user will be displayed "**Find**" pop-up requesting user to enter text to be searched with "**OK**" and "**Cancel**" as the options. When user have entered the text to be searched, he need to click on **OK** button to search the entered text within the script and the searched text would get highlighted and in case user wants to cancel the **Find** operation, user need to click on **Cancel** button.

We also have two other options for finding text i.e. **Find Next(Ctrl+G)** and **Find Previous(Ctrl+Shift+G)**. Suppose user wants to find "**Wait**" text wit in the script and Wait Tag has been used in multiple lines, when user uses **Find** operation, it will highlight searched text, based on the position of cursor and in case, user wants to navigate to the next instance of wait tag he/she needs to either click on **Find Next** option or use shortcut key(**i.e. Ctrl+G**). If user wants to go to previous instance of wait tag, user needs to either click on **Find Previous** option or need to use shortcut key(**i.e.Ctrl+Shift+G**). The screenshot for **Find** operation is provided below:

When user searched particular text:



```

Untitled 13-Apr.xml
81 <!-- TBD : Step description -->
82 <Wait time="2000" />
83
84 <!-- TBD : Step description -->
85 <ClickElement locatorImage="PM_TeIDemo3_Subtype" />
86
87 <!-- TBD : Step description -->
88 <Wait time="2000" />
89
90 <!-- TBD : Step description -->
91 <Action name="rightclick" locatorImage="Physical Product00023804 (Physi" />
92
93 <!-- TBD : Step description -->
94 <Wait time="2000" />
95
96 <!-- TBD : Step description -->
97 <ClickElement locatorImage="PM_TeIDemo3_Subtype" />
98
99 <!-- TBD : Step description -->
100 <Wait time="2000" />

```

- x) **Indent:** When user wants to Indent any line of code, he/she needs to move mouse cursor to that line and need to either click on **Indent** option or by using shortcut key(**i.e.Tab**), then space/Indentation will get added in that line as per mouse cursor position. The Indent settings could be modified from **Preferences** Tab. The screenshot for **Indent** operation is provided below:

Before Indent:

```

Untitled *13-Apr.xml
81 <!-- TBD : Step description -->
82 <Wait time="2000" />
83
84 <!-- TBD : Step description -->
85 <ClickElement locatorImage="PM_TeIDemo3_Subtype" />
86
87 <!-- TBD : Step description -->
88 <Wait time="2000" />
89
90 <!-- TBD : Step description -->
91 <Action name="rightclick" locatorImage="Physical Product00023804 (Physi" />
92

```

After Indent:

```

81      <!-- TBD : Step description -->
82      <Wait time="2000" />
83
84      <!-- TBD : Step description -->
85      <ClickElement locatorImage="PM_TealDemos3_Subpage" />
86
87      <!-- TBD : Step description -->
88      <Wait time="2000" />
89
90      <!-- TBD : Step description -->

```

- xi) **Un-Indent:** When user wants to **Un-Indent** any **Indent** operation, he/she needs to move mouse cursor to that line and need to either click on **Un-Indent** option or by using shortcut key(i.e.**Shift+Tab**), then space/Indentation will get deleted in that line as per mouse cursor position.
- c) **View:** This menu contains operations/options related to XML script image content modification. The options which is present under **View** menu is as follows:
- i) **Toggle Thumbnails:** When user wants to change the view of captured image in image related locators with in the script, he/she needs to either click on **Toggle Thumbnails** option or shortcut key(i.e.**Ctrl+T**). Once user clicks on this option, the image present in the image related locators would get converted from Thumbnails to its name.png(i.e. 15577788333.png) and vice-versa. The screenshot for the same is provided below:

```

81      <!-- TBD : Step description -->
82      <Wait time="2000" />
83
84      <!-- TBD : Step description -->
85      <ClickElement locatorImage="PM_TealDemos3_Subpage" />
86
87      <!-- TBD : Step description -->
88      <Wait time="2000" />
89
90      <!-- TBD : Step description -->
91      <Action name="rightclick" locatorImage="Physical Product00023B04 (Phys)" />
92
93      <!-- TBD : Step description -->
94      <Wait time="2000" />
95
96      <!-- TBD : Step description -->

```

After clicking on **Toggle Thumbnails** option:

```

72      <!-- TBD : Step description -->
73      <ClickElement locatorImage="158e7e1e22219.png" />
74
75      <!-- TBD : Step description -->
76      <Wait time="2000" />
77
78      <!-- TBD : Step description -->
79      <ClickElement locatorImage="158e7e1e46107.png" />
80
81      <!-- TBD : Step description -->
82      <Wait time="2000" />
83
84      <!-- TBD : Step description -->
85      <ClickElement locatorImage="158e7e1e58108.png" />
86
87      <!-- TBD : Step description -->
88      <Wait time="2000" />

```

2. **Record button:** When user wants to create a script with help of Recorder, user needs to click on Recorder button, . Once user clicks on Recorder button, the recording will start immediately. It is recommended to first open your application, then switch to recorder and click on Record button, the recording would start immediately and user could perform different functions, the same would be recorded. Once user stops the script, user would be displayed the script as per their actions performed in **XML Script Editor**. It is also

recommended to first click on new window, which might appear in application in order to record their elements. The Record button image is provided below:



3. **Take screenshot:** When user wants to take screenshot for any element manually in image related attributes(such as locatorImage etc.) within the tag, user need to click on **Take screenshot** button which is located below Toolbar menu and above Tag Library/XML Script Editor. It is recommended to first open your application, then switch to recorder and click on **Take screenshot** button. Once user clicked on **Take screenshot** button, the focus would switch to application displaying message i.e. "**Select an image**". User needs to select the region of element for which he/she needs to capture screenshot. As soon as user finishes marking the region for capturing screenshot, it would immediately appear in XML script editor. The **Take screenshot** button image is provided below:



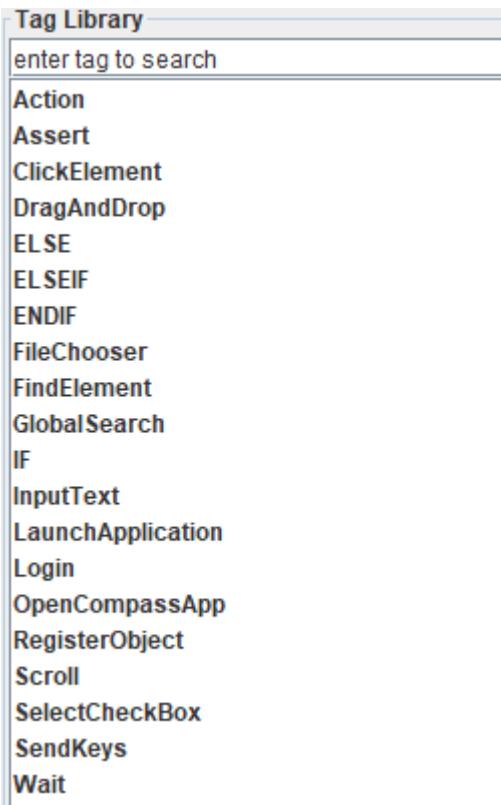
4. **Insert Image:** When user wants to insert screenshot for any element manually in image related attributes(such as locatorImage etc.) within the tag, user need to click on **Insert Image** button which is located below Toolbar menu and above Tag Library/XML Script Editor. It is recommended to first place cursor to the attribute for which he/she needs to insert screenshot. As soon as user clicks on **Insert Image** button, user would be displayed the "**Load Image File**" dialogue box, with **Images** folder of current XML Script, requesting user to select the image which needs to be inserted within the attribute of XML script with "**Select**" and "**Cancel**" as the options. Once the user selects the image and clicks on **Select** button, the image will appear in **XMLScript** immediately. In case user wants to cancel the **Insert Image** operation, he/she needs to click on **Cancel** button , then switch to recorder and click on **Take screenshot** button. The **Take screenshot** button image is provided below:



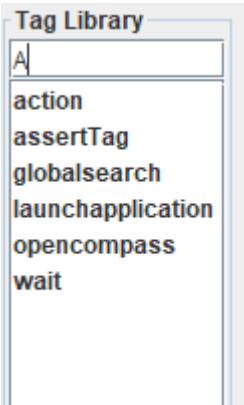
5. **Tag Library:** It is located below **Record** button and before **XML Script Editor** Section and consists list of different tags which could be used with in the 3DX Native App Recorder. The **Tag Library** could be opened by clicking on the **Tag Library** icon, which is displayed in screenshot below:



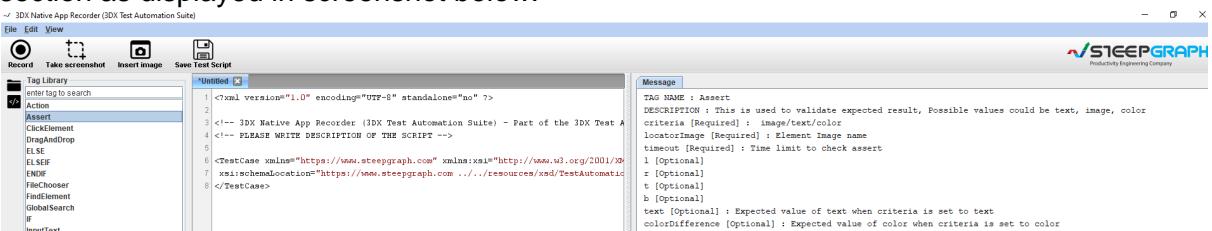
The screenshot for the **Tag Library** is provided below:



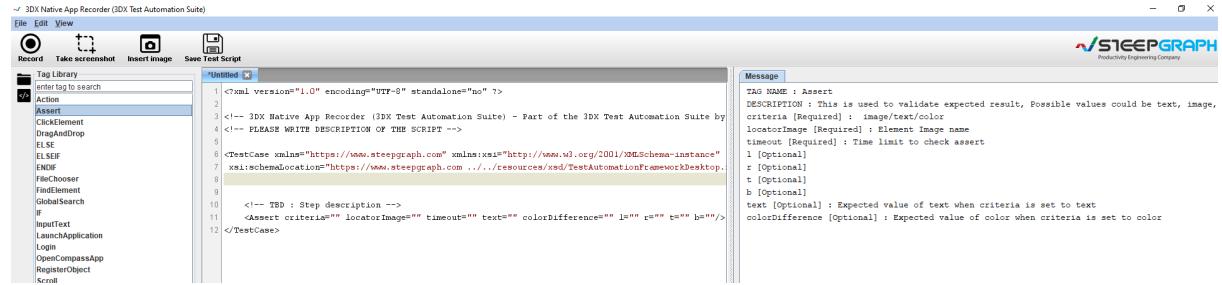
It consists of one search text box displaying text "**enter a tag to search**". User could search the tags from the list of available tags by entering text related to different tags. As soon as user enter the text in it, it would display the tags related to entered text. Suppose user have entered "A" in "**enter a tag to search**" text box, it would display list of tags having entry A in their name at any position i.e. "Action", "asserttag" , "launchapplication" etc. The screenshot for same is provided below:



When user perform single click on any tag from the list of available Tags, the Tag details(such as Tag Name, Tag description and attributes name along with attributes description) would be displayed in the Message section. Suppose user perform single click on **Assert** Tag, the Tag Details related to **Assert** Tag would be displayed in the Message section as displayed in screenshot below:



When user perform double click on any tag from the list of available Tags, the Tag is inserted in the **XML Script Editor** in which user needs to enter the different attributes, in order to perform different actions and operations, along with the Tag Details in Message Section. Suppose user perform double click on **Assert** Tag, the **Assert** Tag would be inserted in the **XML Script Editor** along with the Tag Details in Message Section as displayed in screenshot below:



6. **XML Script Editor:** It is located after the Tag Library and before Message Section. User could write/create/edit scripts in the **XML Script Editor**. User could create scripts in **XML Script Editor** with help of Tag Library and also user could create and edit scripts manually in **XML Script Editor**. The recorded script also appear in same section. The screenshot for same is provided below:



It is recommended for user to keep the structure intact, as in image displayed above.

7. **Message:** In this section, user would be displayed the relevant messages related to the action performed by user related to Tags and creation of script. If the created script is as per standards described for 3DX Native App Recorder, the message section will not display any error message. In case of any issues in scripts, the user would be displayed the related error messages as displayed in the screenshot below:



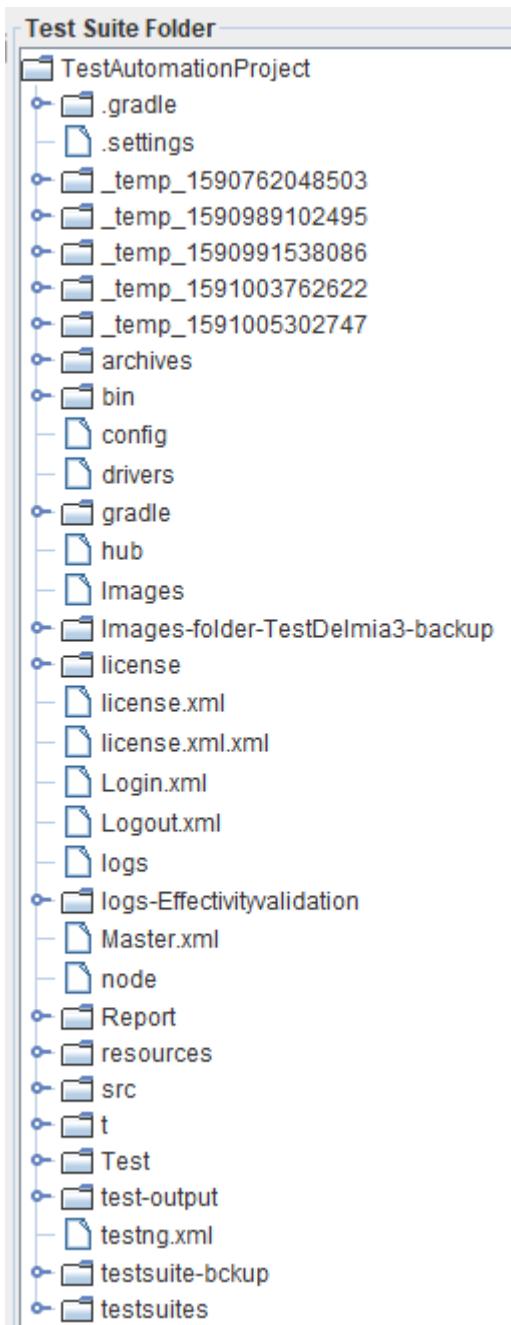
8. **Save Test Script:** When user wants to save the test script at any point of time, user need to click on either **Save Test Script** button which is located below Toolbar menu and above Tag Library/XML Script Editor or by using shortcut key(**i.e. ctrl+s**). Whenever user clicks on **Save Test Script** button, user would be displayed **Input** pop-up asking user to enter desired Test Case name in "**Enter Test Case Name:**" field with "**OK**" and "**Cancel**" as the options. In order to save the Test Script, user needs to enter Test Case Name and click on **OK** button and in order to Cancel the Save operation, user needs to click on **Cancel** button. The **Save Test Script** button image is provided below:



9. **Test Suite Folder/Panel:** It is located above **Tag Library** icon, below **Record** button and before **XML Script Editor** Section. This folder/panel displays the current selected TestSuite folder structure with folder and xml files, which could be expanded and collapsed, as per user actions and requirement. User could open any xml file, by double clicking on any xml file. The xml file would get opened in new Tab. User could open **Test Suite Folder/Panel** by clicking on its icon, which is displayed in screenshot below:



Test Suite Folder Icon



Test Suite Panel

Created with the Personal Edition of HelpNDoc: [Free help authoring environment](#)

Stop_3DX_TAS Browser Instance

Stop.bat: For Kill Instance driver from Task manager

Past this key in Browser.properties file
sg-tas.browser.webdriver.kill.process=msedgedriver.exe

Run this batch file to kill instance driver

Stop_SG_TAS

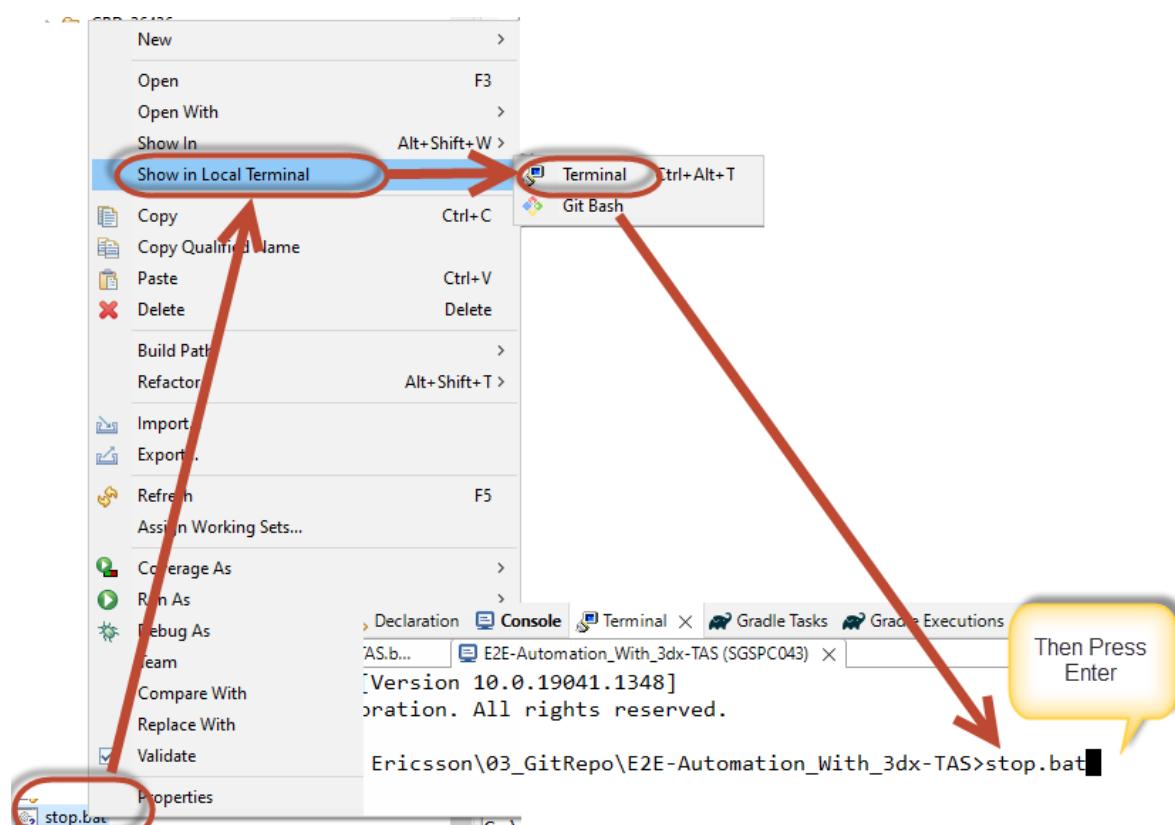
1.) Some Test cases failed if:

- a. We stop execution abruptly.
- b. We forget to close the driver from task manager.
- c. One or more instance already running in task manager.
- d. Takes more time to kill instance driver.

Solution: we reduced this process by executing stop.bat file in terminal

1) Stop batch file is used to help kill driver for

- a. To kill multiple instance driver at a time from task manager.
- b. No need to kill instance driver separately from task manager.
- c. Only Execute stop.bat file from terminal or double clicking on it,
- d. It simple to use stop.bat from terminal.
- e. Then it will automatically kill all instance driver from task manager.



Created with the Personal Edition of HelpNDoc: [Easily convert your WinHelp HLP help files to CHM with HelpNDoc's step-by-step guide](#)

Auto Browsers Driver Download

3DX_TAS Auto Browser Download :

```
sg-tas.browser.webdriver.autodownload=true
```

We don't need to download latest driver instance for Manually only Above property needs to be added in Browser.properties file to use this feature and it will automatically downloaded in your drivers folder

This is use for ChromeDriver, FirefoxDriver, EdgeDriver, OperaDriver, InternetExplorerDriver, etc

Created with the Personal Edition of HelpNDoc: [Maximize Your Documentation Capabilities with a Help Authoring Tool](#)

3DX Dataloader

- **Introduction:**

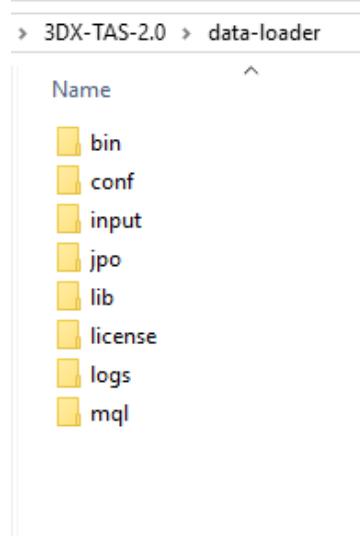
Data Loader is a application used for bulk import or delete of data. This document describes the steps for using Data Loader (i.e. its configuration detail, execution detail etc.).

- **Execution Steps:**

- **Data Loading:**

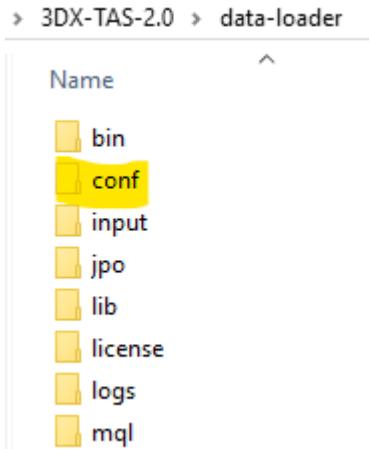
- **Copy:**

Navigate to data-loader folder to your SG-TAS build and copy the same. The folder structure of data-loader is provided below in screenshot:



▪ Configuration:

- (1) Navigate to **conf** folder as displayed in screenshot provided below:



- (2) Update **DataLoader.properties** file, by updating **MQLMaster.port** to 6788, as displayed in the screenshot provided below:

```
## Generate MQL ##
FStoMQL.FS.StartRow = 4

## Execute MQL ##
#-----
#Server Details
MQLMaster.host=localhost
MQLMaster.port=6788

#User Details:
#-----
EMX_HOST=http://192.168.0.90:8070/internal
EMX_USER=creator
EMX_PASSWORD=
EMX_ROLE=ctx:Senior Manufacturing Engineer.STRATA.STRATA_Internal

# Maximum Threads
# Default 1
#-----
MQL.ThreadCount=1

# Static Data Load for WPs
#-----
Static.WORK_PACKAGE_NAME=

# WORK_PACKAGE_NAME: Work package folder Name
#-----

Prerequisite.WORK_PACKAGE_NAME=ERI_Rel_Connect
#Prerequisite.WORK_PACKAGE_NAME=BOMManagement,CALifeCycle,NCDataManagement,Andon,IssueManagement,SearchModule,EngineeringSpecification,ProgramManagement
,Specifications,Document
#04_WP03B1,05_WP03B2,12_WP11
```

- (3) **FStoMQL.FS.StartRow:** This attribute provides the line number of functional sheet, from where user/we wants to start the MQL creation:
For example: FStoMQL.FS.StartRow=4

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
Domain	TEST CASE TYPE	NAME	REVISION POLICY	STATE	ENOVI A Attribute	Attribute	Direction	ENOVI A Functions TYPE	NAME	REVISION POLICY	STATE	ENOVI R Rel	Attrib	Relations	Level						
1	2	3	4	PART	123																
5																					
6																					
7																					
8																					
9																					
10																					
11																					
12																					
13																					
14																					
15																					
16																					
17																					
18																					
19																					

- (4) **Master Server Details:** In this, user needs to add host and port detail of **MQLmaster** as displayed in screenshot provided below:

```
#-----
## Execute MQL ##
#-----
#Server Details
MQLMaster.host=localhost
MQLMaster.port=6788
```

- (5) **User Details:** In this user needs to provide Server URL details along with User Name, Password and User Role, as displayed in screenshot provided below:

```
#User Details:
#-----
EMX_HOST=http://192.168.0.90:8070/internal
EMX_USER=creator
EMX_PASSWORD=
EMX_ROLE=ctx::Senior Manufacturing Engineer.STRATA.STRATA_Internal
```

- (6) **MQL Thread Count:** In this, user needs to provide the number of thread that user wants to run. The default value of thread count is 1. The screenshot for the same is provided below:

```
21
22 # Maximum Threads
23 # Default 1
24 #-----
25
26 MQL.ThreadCount=1
27
```

- (7) User also needs to provide name of **work packages**, user wants to execute in, **dataloader.properties** file.

For example: In our case, we are going to execute **ERI_Rel_Connect** package. The screenshot for same is provided below:

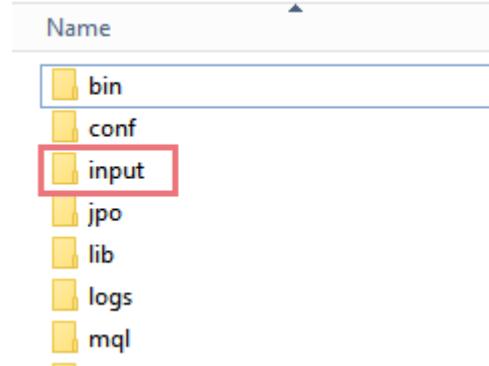
```
28 # static Data Load for WPs
29 <!--
30 Static:WORK_PACKAGE_NAME=
31 <!--
32 # WORK_PACKAGE_NAME: work package folder Name
33 <!--
34 Prerequisite:WORK_PACKAGE_NAME=ERI_Rel_Connect
35 #Prerequisite:WORK_PACKAGE_NAME=ECManagement,CALifeCycle,NCDataManagement,Andon,IssueManagement,SearchModule,EngineeringSpecification,ProgramManagement
,Specifications,Document
36 <!--
37 #04_WP03B1,05_WP03B2,12_WP11
38 <!--
```

- (8) **Update Input.properties file:**

This file provides input to Data Handler to generate the functional Sheet. In this, user needs to map the columns to the respective **KEY** to be used by Handler. We need to add details regarding creation of functional sheet inside **Input.properties**, as displayed in the screenshot provided below along with naming conventions used:

Conventions used:

<INPUT_FILE_NAME> - This refers to the excel file name present under "input" folder.



<INPUT_SHEET_NAME> - This refers to the sheet name present inside **<INPUT_FILE_NAME>**.



<INPUT_HANDLER_NAME> - This refers to JAVA class name to handle the input sheet.

(If specific handler is provided, then user need to add handler name).

```

13 #####
14
15
16 # Properties for Excel file
17 -----
18
19 # Properties for Excel Sheets
20 # Sheet will not be processed unless the mapping of sheet is available
21 -----
22
23 # Name of sheets to be processed
24 <INPUT_FILE_NAME>.ProcessSheets=All
25
26 # Data start row of Sheet
27 <INPUT_SHEET_NAME>.RowStart=1
28
29 # Java Handler name for sheet
30 <INPUT_SHEET_NAME>.GenerateFunctionalData.ClassName=<INPUT_HANDLER_NAME>
31
32 # Split the data into multiple Files in case of big data
33 <INPUT_SHEET_NAME>.SplitFunctionalSheet=Yes
34 <INPUT_SHEET_NAME>.SplitFunctionalSheet.SplitByLines=20000
35
36 # Sheet Name of output excel
37 <INPUT_SHEET_NAME>.GenerateFunctionalData.NewSheetName=PrerequisiteData
38
39 # For Generic Handler (Note : <INPUT_SHEET_NAME>.GenerateFunctionalData.ClassName must be blank )
40 <INPUT_SHEET_NAME>.FunctionalColumns.InputMappingExcel=ERI_Rel_Connect.xlsx
41 <INPUT_SHEET_NAME>.GenerateWithoutMapping=True
42

```

<INPUT_FILE_NAME>.ProcessSheets : The value for this attribute should be "ALL" (It is recommended to take all file to generate FS)

<INPUT_FILE_NAME>.RowStart : In this, user needs to provide sheet line number from which functional sheet generation starts.

<INPUT_SHEET_NAME>.GenerateFunctionalData.ClassName : In this user should add handler, according to functional sheet used for mapping.

- **InputMapping:**

- Go to input folder : Input folder contains 2 Excel(in .xlsx extension) file,in which 1st file contains input data and another file called **InputMappings.xlsx**, which is used for mapping unique keys with column header of 1st Excel(in .xlsx extension) file.

Unique Keys are displayed in screenshot provided below:

A	B	C	D
Identifiers:			
	LEVEL_ID		
	FROM_TYPE		
	FROM_NAME		
	FROM_REVISION		
	FROM_POLICY		
	FROM_STATE		
	FROM_ATTR		
	TO_RELATIONSHIP		
	TO_TYPE		
	TO_NAME		
	TO_REVISION		
	TO_POLICY		
	TO_STATE		
	TO_ATTR		
	REL_ATTR		
	LEVEL_REL		
	LEVEL_REL_ATTR		

For example:

A	B	C	E
Sr. No.	Column Name	Identifier	Default
1	1 Level	LEVEL_ID	
2	2	FROM_TYPE	PART
3	3	FROM_NAME	123
4	4 FromRevision	FROM_REVISION	-
5	5 RelName	TO_RELATIONSHIP	Product
6	6	FROM_RELATIONSHIP	Product
7	7	FROM_ATTR	
8	8 Attr1,Attr2	REL_ATTR	
9	9 ToType	TO_TYPE	
10	10 ToName	TO_NAME	
11	11 ToRevision	TO_REVISION	
12			
13			
14			
15			
16			
17			
18			
19			
20			
21			
22			
23			
24			
25			
26			
27			
28			
29			
30			
31			
32			
33			
34			
35			
36			
37			
38			
39			
40			

ERI_Rel_Connect

InputMappings

In above figure, **Column Name**: contains all column header of **ERI_REL_Connect** file

Identifier: contains all Unique keys

Default : contains default value of Key (When **ERI_REL_Connect**

file contains blank value, then it will take default value), as displayed in screenshot provided below:

A	B	C	D	E	F	G	H	I
1	FromType	FromName	FromRevision	RelName	Attr1	Attr2	ToType	ToName
2	Product Line	MOA Object 009	-	Product Line Models	Val2 for 005	Val2 for 009	ERI_MarketOffering	MO Object 005
3		MOA Object 009	-	Product Line Models	Val2 for 004	Val2 for 006	ERI_MarketOffering	MO Object 006
4	Product Line	MOA Object 007	-	Product Line Models	Val2 for 003	Val2 for 007	ERI_MarketOffering	MO Object 007
5	Product Line	MOA Object 008	-	Product Line Models	Val2 for 002	Val2 for 008	ERI_MarketOffering	MO Object 008
6								
7								
8								
9								
10								
11								
12								
13								
14								
15								
16								
17								
18								

Contain blank value so it will contain default value i.e Part

We could also add multiple column to input mapping sheet for attributes as displayed in screenshot provided below. It would take value from column name with key as **REL_ATTR**.

REL_ATTR: Attribute present on relationship.

If attribute column is not available, then there would be no value in functional sheet for that attribute.

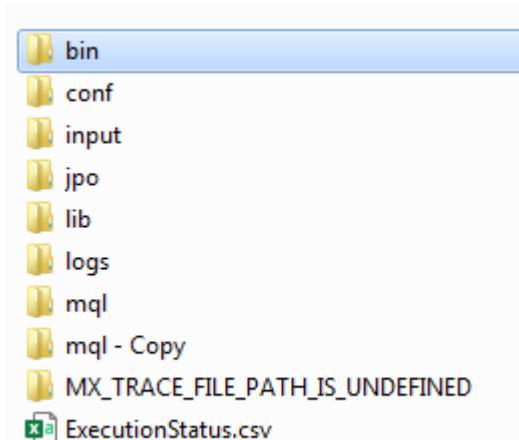
For example: For FROM_ATTR, there is no column name available at parent side, so attribute value would appear as **BLANK**, as displayed in screenshot provided below:

A	B		C	E
1	Sr. No.	Column Name	Identifier	Default
2	1	Level	LEVEL_ID	
3	2		FROM_TYPE	PART
4	3		FROM_NAME	123
5	4	FromRevision	FROM_REVISION	-
6	5	RelName	TO_RELATIONSHIP	Product
7	6		FROM_RELATIONSHIP	Product
8	7		FROM_ATTR	
9	8	Attr1,Attr2	REL_ATTR	
10	9	ToType	TO_TYPE	
11	10	ToName	TO_NAME	
12	11	ToRevision	TO_REVISION	
13				

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	Domain	TEST CASE TYPE	NAME	REVISION POLICY	STATE	ENOVIAS Attribute Actual Name	Attribute Value	Direction	ENOVIAS Function Type	NAME	REVISION POLICY	STATE	ENOVIAS Rel Attr Relations Lev							
2		PART	123	-				o	Product Li	ERI_Market MO Objec-										
3								o												
4		PART	123	-				o												
5								o												
6								o												
7								o												
8								o												
9								o												
10								o												
11								o												
12								o												
13								o												
14								o												
15								o												

- Execution:

Navigate to bin folder:

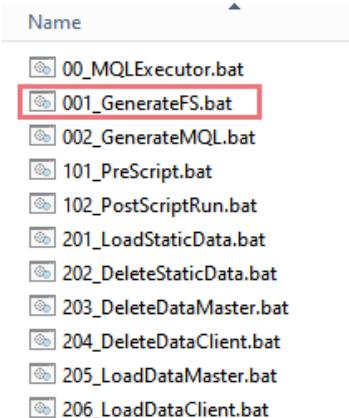


NOTE: It is recommended to clear logs before execution.

User should run the files in following sequence:

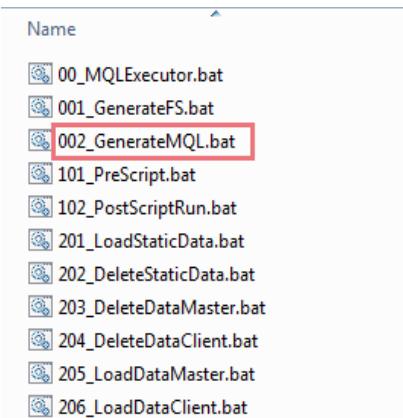
(1) **Generate Functional Sheet:**

001_GenerateFS.bat: This will generate Functional sheet and for the same, we have to configure **Input.properties** file



(2) **Generate MQL Files:**

002_GenertateMQL.bat: This would generate MQL for the packages that we have configured in the **DataLoader.properties** file.

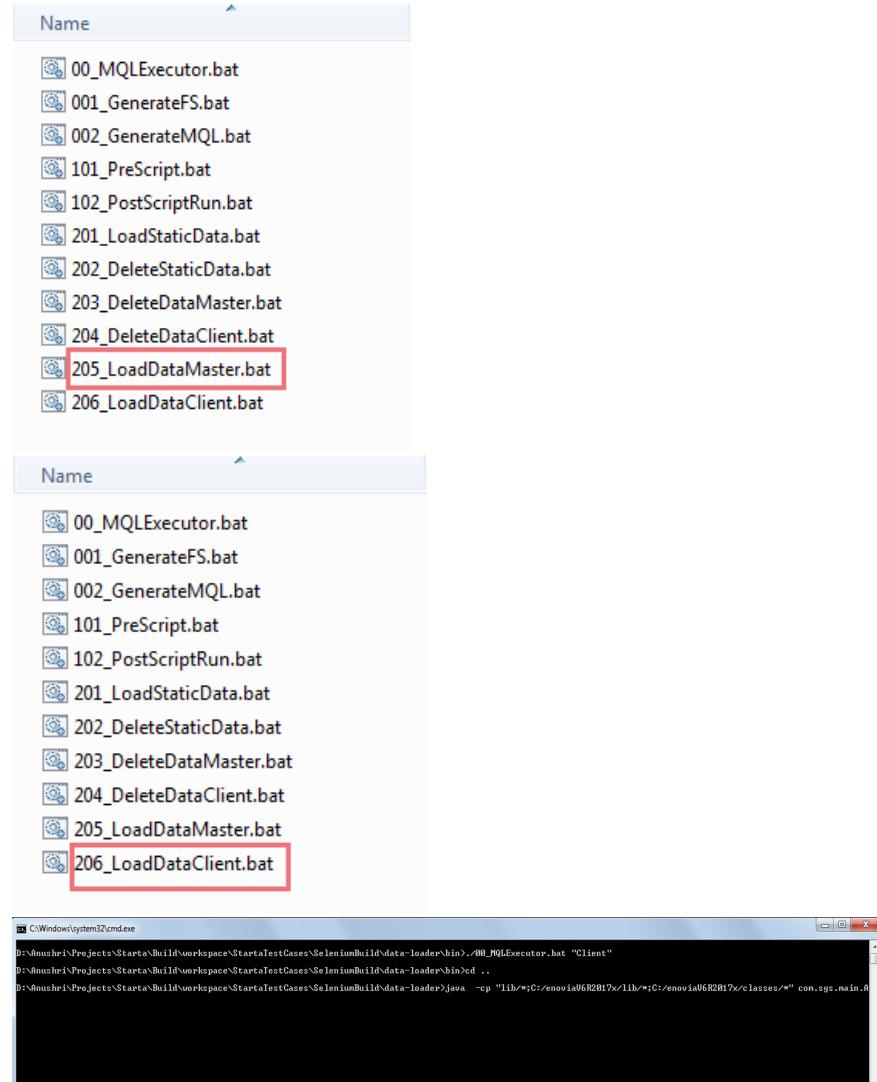


User could find the logs in the **logs** folder

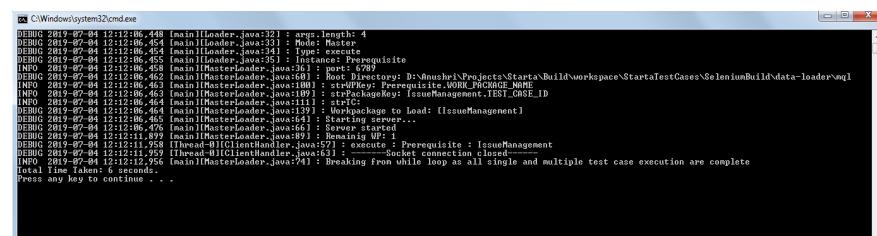
(3) **Load Data:**

Run **205_LoadDataMaster.bat** and then

206_LoadDataClient.bat.

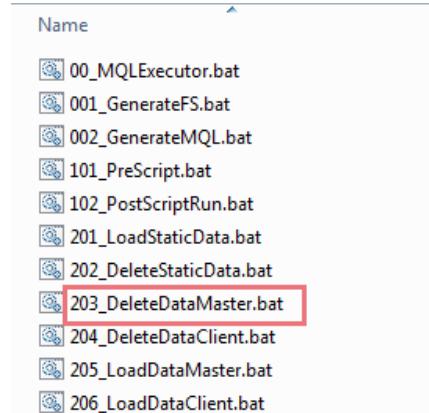


After execution of **206_LoadDataClient**. You would see following message on **Master.bat** window.



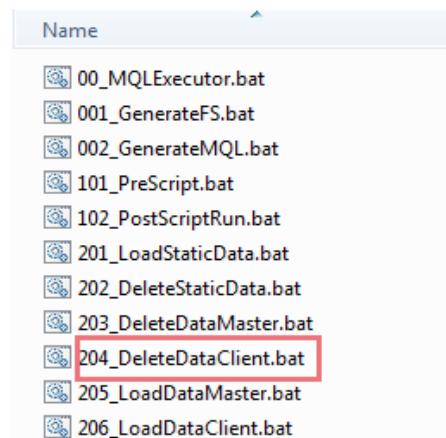
(4) Delete Data:

Run **203_DeleteDataMaster.bat** and then
204_DeleteDataClient.bat to delete existing data.



```
C:\Windows\system32\cmd.exe
DEBUG 2019-07-04 11:53:14.098 [main][Loader.java:32] : args.length: 4
DEBUG 2019-07-04 11:53:14.101 [main][Loader.java:33] : Mode: Master
DEBUG 2019-07-04 11:53:14.102 [main][Loader.java:34] : Type: delete
DEBUG 2019-07-04 11:53:14.102 [main][Loader.java:35] : Prerequisites:
INFO 2019-07-04 11:53:14.104 [main][MasterLoader.java:36] : port: 6789
DEBUG 2019-07-04 11:53:14.106 [main][MasterLoader.java:60] : Root Directory: D:\nushri\Projects\Starta\Build\workspace\StartaTestCases\SeleniumBuild\data-loader\nql
INFO 2019-07-04 11:53:14.106 [main][MasterLoader.java:109] : strPackageKey: IssueManagement.TEST_CLOSE_ID
INFO 2019-07-04 11:53:14.106 [main][MasterLoader.java:113] : strPkg:
INFO 2019-07-04 11:53:14.106 [main][MasterLoader.java:113] : package to Load: {IssueManagement}
DEBUG 2019-07-04 11:53:14.107 [main][MasterLoader.java:64] : Starting server...
DEBUG 2019-07-04 11:53:14.115 [main][MasterLoader.java:66] : Server started
```

This is master file, keep this file running and then run **DeleteDataClient.bat**.



```
C:\Windows\system32\cmd.exe
D:\nushri\Projects\Starta\Build\workspace\StartaTestCases\SeleniumBuild\data-loader\bin>./00_MQLExecutor.bat "Client"
D:\nushri\Projects\Starta\Build\workspace\StartaTestCases\SeleniumBuild\data-loader\bin>d ..
D:\nushri\Projects\Starta\Build\workspace\StartaTestCases\SeleniumBuild\data-loader>java -cp "lib/*;C:/enviav6R281?/lib/*;C:/enviav6R281?/classes/*" con.sgs.main
```

DeleteDataClient.bat file window would close after execution.
Now come back to **DeleteDataMaster** window. You would see following message on the screen. Now we could close the **DeleteDataMaster** window or Press any key to continue.

```
C:\Windows\system32\cmd.exe
DEBUG 2019-07-04 11:53:14.098 [main][Loader.java:32] : args.length: 4
DEBUG 2019-07-04 11:53:14.101 [main][Loader.java:33] : Mode: Master
DEBUG 2019-07-04 11:53:14.102 [main][Loader.java:34] : Type: delete
DEBUG 2019-07-04 11:53:14.102 [main][Loader.java:35] : Prerequisites:
INFO 2019-07-04 11:53:14.104 [main][MasterLoader.java:36] : port: 6789
DEBUG 2019-07-04 11:53:14.106 [main][MasterLoader.java:60] : Root Directory: D:\nushri\Projects\Starta\Build\workspace\StartaTestCases\SeleniumBuild\data-loader\nql
INFO 2019-07-04 11:53:14.106 [main][MasterLoader.java:109] : strPackageKey: IssueManagement.TEST_CLOSE_ID
INFO 2019-07-04 11:53:14.106 [main][MasterLoader.java:113] : strPkg:
INFO 2019-07-04 11:53:14.106 [main][MasterLoader.java:113] : package to Load: {IssueManagement}
DEBUG 2019-07-04 11:53:14.107 [main][MasterLoader.java:64] : Starting server...
DEBUG 2019-07-04 11:53:14.115 [main][MasterLoader.java:66] : Server started
DEBUG 2019-07-04 11:58:03.655 [main][MasterLoader.java:89] : Remaining WP: 1
DEBUG 2019-07-04 11:58:03.740 [Thread-0][ClientHandler.java:57] : delete : Prerequisites : IssueManagement
INFO 2019-07-04 11:58:04.722 [main][MasterLoader.java:74] : Breaking from while loop as all single and multiple test case execution are complete
Total time taken: 4 minutes 50 seconds.
Press any key to continue . . .
```

Created with the Personal Edition of HelpNDoc: [Maximize Your PDF Protection with These Simple Steps](#)

Write Test Scripts

Created with the Personal Edition of HelpNDoc: [Effortlessly upgrade your WinHelp HLP help files to CHM with HelpNDoc](#)

How to write Test Script/TestCase

1. Open the text editor of your choice.

NOTE: Notepad++ or Eclipse XML Editor is a recommended/common choice, but other text editors can also be used to create XML files.

2. Type the following lines at the beginning of the document:

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
```

3. To access the XML schema, place the schema-dependent paths in given fragment:

```
<TestCase xmlns="https://www.steepgraph.com"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="https://www.steepgraph.com ../../
  ..../resources/xsd/TestAutomationFramework.xsd" targetApplication="web">
```

The following fragment:

xmlns="https://www.steepgraph.com" - is used to declare that all the elements used in this XML document are declared in the "https://www.steepgraph.com" namespace.

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" - indicates that the elements and data types used in the schema come from the "http://www.w3.org/2001/XMLSchema-instance" namespace.

**xsi:schemaLocation="https://www.steepgraph.com ../../
/resources/xsd/TestAutomationFramework.xsd"** - is used to specify schema Location.

This attribute has two values, separated by a space. The first value is the name space to be used. The second value is the location of the XML schema, to be used for that namespace. The three dots with backslash, specify the path in which the xsd file, which is available.

targetApplication is a optional attribute to specify that test script/test case is for web or desktop/native. Possible values are 'web' and 'desktop'. If not specified then default is web

4. **Test script writing format :** complex type = ComplexType (first letter to be capital for each word) [TAG NAME]

Simple type === simpleType (camel case) [ATTRIBUTE]
 Value / enumeration === value_enumeration (all letters to be small) [VALUE]

5. Now type the contents of your block.

For available list of tags, see TAG </> Library section.

- **Syntax:**

```
<InputText locatorType="id" locatorExpression="Name" id="PartName"/>
```

- **Attributes:**

- **LocatorType:**

There are 4 types of locator as mentioned below:

- **id:**
This attribute is used to find the element on web page, using id attribute of web element.
- **cssselector:**
This attribute is used to find element on web page, using css expression.
- **name:**
This attribute is used to find element on web page, using name attribute of web element expression.
- **xpath:**
This attribute is used to find element on web page, using xpath expression.
- **locatorExpression:**
This attribute contains expression, specific to locator type. If locatorType is id, then locatorExpression should contain id of the web element or if locatorType is xpath then xpath expression should be given in this attribute.
- **id:**
Id attribute is used to take value, used to search from csv. id attribute is already described in InputText tag.

Example:

Support user like to input A-0000104-01 Part name on Create Part form. First, he/she needs to define column in csv and enter its value as "A-0000104-01".

▪ **Example:**

```
<ClickElement locatorType="id" locatorExpression="DescriptionId"/>
<ClickElement locatorType="xpath"
locatorExpression="//textarea[@id='DescriptionId']"/>
```

6. Include the closing tag at the end of the document:

</TestCase>

7. Save this new file on your hard drive, with an .xml extension (for example, CreatePart.xml).

8. Code Sample:

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?><TestCase
xmlns="https://www.steepgraph.com" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:schemaLocation="https://www.steepgraph.com ../../
..//resources/main/TestAutomationFramework.xsd">
<SwitchToDefaultContent />
<Wait time="3000" />
<ClickGlobalActionsMenu commandLabel="Engineering|Part|Create Part"/>
<SwitchToSlideInWindow />
<ClickElement locatorType="xpath"
locatorExpression="//table[@class='form']//input[@name='autoNameCheck']" />
<SelectElement locatorType="id" locatorExpression="AutoNameSeriesId"
id="AutoName_Series"/>
<FindElement locatorType="xpath" locatorExpression="//button[@class='btn-primary']"
id="Ok"/>
<Action name="click" refid="Ok"/>
```

</TestCase>

Created with the Personal Edition of HelpNDoc: Write EPub books for the iPad

How to use xsd suggestions from eclipse

1. You can launch content assist in either of the following ways:
From the Edit menu, click Content Assist, or Press Ctrl+Space
2. In addition, you can set up an option, that causes content assist to pop up automatically, when certain characters are typed. To set up this option, click Window >Preferences to open the Preferences window, then select XML > XML Files > Editor > Content assist. Select Automatically make suggestions check box, and supply any additional characters that should trigger content assist.
3. As you type the first one or two letters of the tag that you want, the list automatically refreshes with alphabetized choices that match your input. Scroll down and select the tag that you want to use by double-clicking on it.

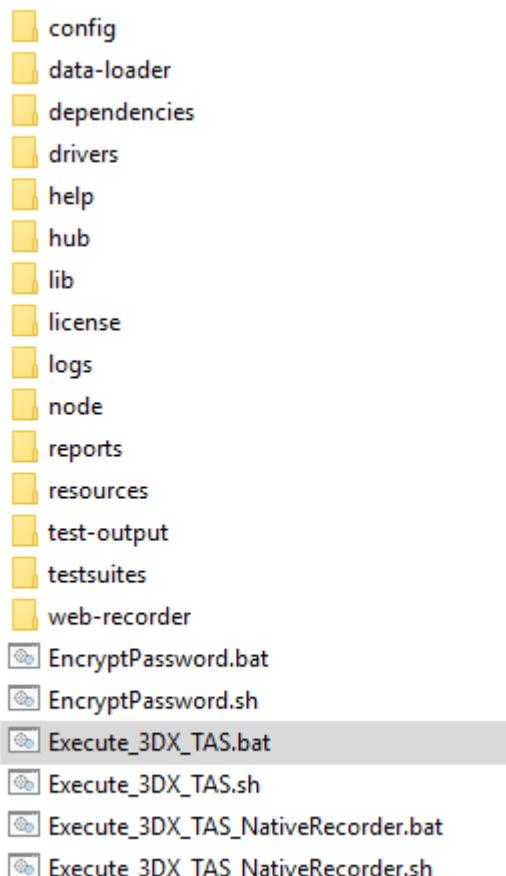
Note:

The list only refreshes as described if you first type < before prompting for content assist.

Created with the Personal Edition of HelpNDoc: [Benefits of a Help Authoring Tool](#)

Execute

1. Go to 3DX-TAS folder, and double on **Execute_3DX_TAS.bat** file or right click on **Execute_3DX_TAS.bat** and open **Terminal**, by clicking on **Terminal option**, from "Show in Local Terminal" menu. Once user is navigated to **Terminal**, user needs to enter command "**Execute_3DX_TAS.bat**" in Terminal, in order to start the Execution



- After executing above command,it should execute the test case that,user has specified in TestPlan.xml. & in cmd it will show the Test Case with the Tag which is being executed on browser.

```
C:\WINDOWS\system32\cmd.exe
N3DX-TAS execution started....!
[TestNG] Running:
D:\Anjali\NEW PROJECT(26-08-2019)\ScenarioTracker\TestAutomationbuild-1.1\testsuites\UC01\UC01.xml

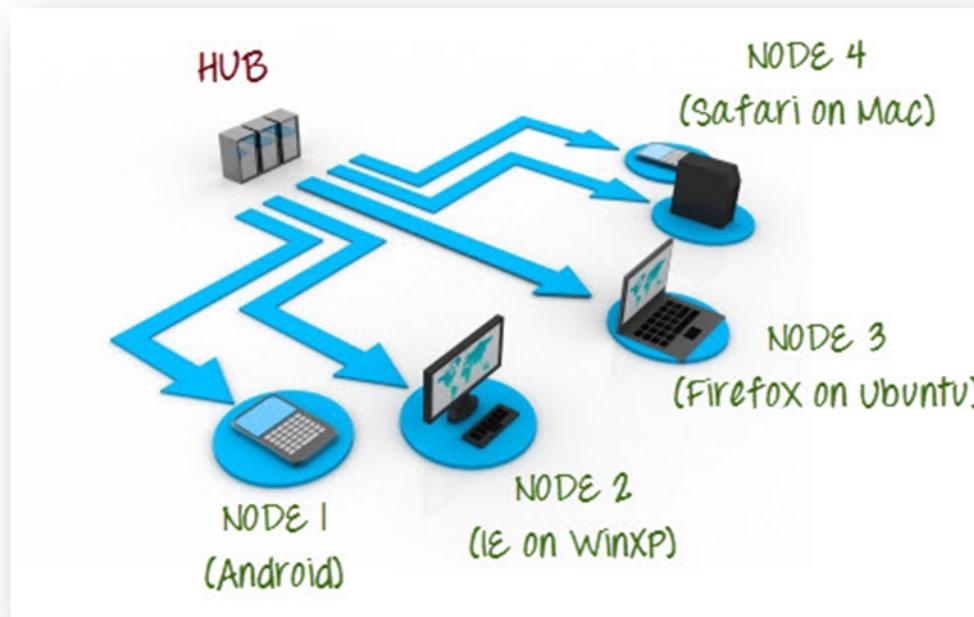
[Ljava.lang.Object;@72d1ad2e
Started InternetExplorerDriver server (32-bit)
3.14.0.0
Listening on port 37182
Log level is set to ERROR
Only local connections are allowed
Mar 16, 2020 2:51:09 PM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: W3C
Test Case XML: testsuites\UC01\TC00\PL.xml, Suite: UC01, Test Case: PL, Status: In Progress
PL.xml => UC01 => PL => <Login> Processing
```

Created with the Personal Edition of HelpNDoc: [Produce electronic books easily](#)

Parallel Execution

Parallel execution of same Test Case on same or different Machines

3DX-TAS uses the Selenium Grid concept where you only run the test on a single machine called a **hub**, but the execution will be done by different machines called **nodes**.



Setting a Hub

Start a Hub using below command line : e.g. **java -jar selenium-server-standalone-2.30.0.jar -role hub**

Setting a Node

Start a Node using below command line: **java -Dwebdriver.<xx>.driver=<D:\drivers\xx.exe> -jar selenium-server-standalone-2.30.0.jar -role webdriver -hub <HubURL> -port <5556>**

Launching a Test Case

Start Hub
Start Node
Launch the Test case execution

How to do it in 3DX-TAS

To Activate Selenium Grid Approach

Set **browser.remote** property as **true** in **Browser.properties** file
3dx-tas.browser.remote=true

Update suite file/test plan file path in 3DX_TAS.properties as below

3dx-tas.suitefile=Master.xml

e.g.

Master.xml file should include your suite of suites file as below

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">
<suite name="Master">
<suite-files>
<suite-file path="testsuites\\master\\Login.xml"></suite-file>
<suite-file path="testsuites\\master\\Login2.xml"></suite-file>
</suite-files>
</suite>
```

Login.xml includes

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">
<suite name="Login">
<test name="Test-Web">
<parameter name="xmlFilePath" value="testsuites\\Test-Web\\Test-Web.xml"></parameter>
<parameter name="csvFilePath" value="testsuites\\Test-Web\\Test-Web.csv"></parameter>
<classes>
<class name="com.steepgraph.ta.framework.App" />
</classes>
</test> <!-- Master -->
</suite> <!-- Suite -->
```

Login2.xml includes

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">
```

```

<suite name="Login2">
<test name="Test-Web1">
<parameter name="xmlFilePath" value="testsuites\\Test-Web1\\Test-Web1.xml"></parameter>
<parameter name="csvFilePath" value="testsuites\\Test-Web1\\Test-Web1.csv"></parameter>
<classes>
<class name="com.steepgraph.ta.framework.App" />
</classes>
</test> <!-- Master -->
</suite> <!-- Suite -->

```

Start a Hub:

Run **StartHub.bat** from the following path: **3DX-TAS\hub\StartHub.bat**

File content:

```

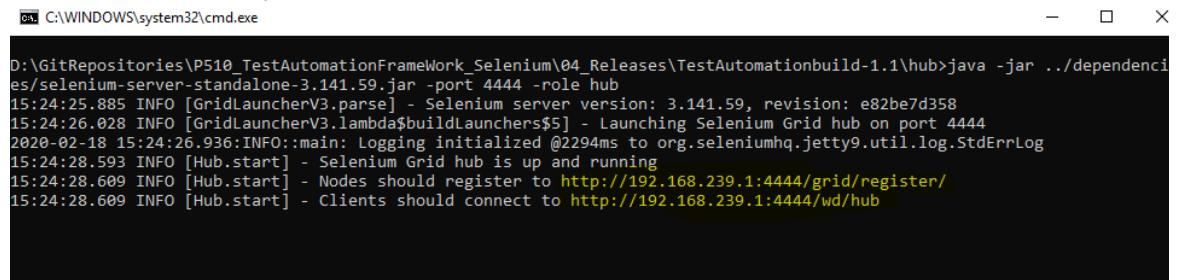
@echo off
for %%B in (%~dp0\.) do set PARENTDIR=%%~dpB

set SELENIUM_SERVER_JAR=%PARENTDIR%dependencies\selenium-server-standalone-3.141.59.jar
set HUB_PORT="4444"

java -jar %SELENIUM_SERVER_JAR% -port %HUB_PORT% -role hub

```

You can change the HUB_PORT if port 4444 is already in use

Hub Executor Prompt:


```

D:\GitRepositories\P510_TestAutomationFramework_Selenium\04_Releases\TestAutomationbuild-1.1\hub>java -jar ..\dependencies\selenium-server-standalone-3.141.59.jar -port 4444 -role hub
15:24:25.885 INFO [GridLauncherV3.parse] - Selenium server version: 3.141.59, revision: e82be7d358
15:24:26.028 INFO [GridLauncherV3.lambda$buildLaunchers$5] - Launching Selenium Grid hub on port 4444
2020-02-18 15:24:26.936:INFO::main: Logging initialized @2294ms to org.seleniumhq.jetty9.util.log.StderrLog
15:24:28.593 INFO [Hub.start] - Selenium Grid hub is up and running
15:24:28.609 INFO [Hub.start] - Nodes should register to http://192.168.239.1:4444/grid/register/
15:24:28.609 INFO [Hub.start] - Clients should connect to http://192.168.239.1:4444/wd/hub

```

Start a Node:

Run **StartNode.bat** from the following path: **3DX-TAS\node\StartNode.bat**

File content:

```

@echo off
for %%B in (%~dp0\.) do set PARENTDIR=%%~dpB
set DRIVER_CHROME=%PARENTDIR%drivers\chromedriver-81.0.4044.69.exe"
set DRIVER_IE=%PARENTDIR%drivers\IEDriverServer_Win32_3.150.1.exe"
set DRIVER_FIREFOX=%PARENTDIR%drivers\geckodriver-v0.26.0-win32.exe"
set SELENIUM_SERVER_JAR=%PARENTDIR%dependencies\selenium-server-standalone-3.141.59.jar"
set HUB_REGISTER_URL="http://192.168.233.1:4444/grid/register"
set NODE_PORT="5577"

java -Dwebdriver.chrome.driver="%DRIVER_CHROME%" -Dwebdriver.ie.driver="%DRIVER_IE%" -Dwebdriver.gecko.driver="%DRIVER_FIREFOX%" -jar %SELENIUM_SERVER_JAR% -role node -hub %HUB_REGISTER_URL% -port %NODE_PORT%

```

Update the HUB_REGISTER_URL with valid ip address and port (it will be available at hub console in previous step) ,NODE_PORT and DRIVER versions if needed.

Node Executor Prompt:

```
D:\GitRepositories\PS10_TestAutomationFramework_Selenium\04_Releases\TestAutomationbuild-1.1\node>java -Dwebdriver.chrome.driver="D:\ProjectWork\TestAutomationSuite\chromedriver_win32\chromedriver.exe" -jar ..\dependencies\selenium-server-standalone-3.141.59.jar -role webdriver -hub http://192.168.239.1:4444/grid/register/ -port 5556
15:35:42.644 INFO [GridLauncherV3.parse] - Selenium server version: 3.141.59, revision: e82be7d358
15:35:42.740 INFO [GridLauncherV3.lambda$buildLaunchers$7] - Launching a Selenium Grid node on port 5556
2020-02-18 15:35:43.138:INFO::main: Logging initialized @704ms to org.seleniumhq.jetty9.util.log.StdErrLog
15:35:43.306 INFO [WebDriverServlet.<init>] - Initialising WebDriverServlet
15:35:43.369 INFO [SeleniumServer.boot] - Selenium Server is up and running on port 5556
15:35:43.369 INFO [GridLauncherV3.lambda$buildLaunchers$7] - Selenium Grid node is up and ready to register to the hub
15:35:43.501 INFO [SelfRegisteringRemote$1.run] - Starting auto registration thread. Will try to register every 5000 ms.
15:35:43.993 INFO [SelfRegisteringRemote.registerToHub] - Registering the node to the hub: http://192.168.239.1:4444/grid/register
15:35:44.180 INFO [SelfRegisteringRemote.registerToHub] - The node is registered to the hub and ready to use
```

Hub Executor Prompt after Node is registered to Hub:

```
D:\GitRepositories\PS10_TestAutomationFramework_Selenium\04_Releases\TestAutomationbuild-1.1\hub>java -jar ..\dependencies\selenium-server-standalone-3.141.59.jar -port 4444 -role hub
15:24:25.885 INFO [GridLauncherV3.parse] - Selenium server version: 3.141.59, revision: e82be7d358
15:24:26.028 INFO [GridLauncherV3.lambda$buildLaunchers$5] - Launching Selenium Grid hub on port 4444
2020-02-18 15:24:26.936:INFO::main: Logging initialized @2294ms to org.seleniumhq.jetty9.util.log.StdErrLog
15:24:28.593 INFO [Hub.start] - Selenium Grid hub is up and running
15:24:28.609 INFO [Hub.start] - Nodes should register to http://192.168.239.1:4444/grid/register/
15:24:28.609 INFO [Hub.start] - Clients should connect to http://192.168.239.1:4444/wd/hub
15:35:44.180 INFO [DefaultGridRegistry.add] - Registered a node http://192.168.239.1:5556
```

Now copy this Node URL from Hub console e.g. "<http://192.168.239.1:5556>" in **Browser.properties** file and update below key
browser.remote.nodeurl=<http://192.168.239.1:5556>/wd/hub

After all above steps start execution of Test Suite/Test Plan using **Execute_3DX_TAS.bat**

Parallel Execution of Test cases

3DX-TAS uses the TestNG approach for parallel execution of Test cases.

Parallel Execution using Selenium Grid and TestNG framework

Activate below properties in **3DX_TAS.properties** file. For parallel execution, use suite-of-suites file as Master suite file.

Set **3dx-tas.suitefile=Master.xml** and **3dx-tas.parallel.execution.poolsize** size as required e.g. **5**

After above steps start execution of Test Suite/Test Plan using **Execute_3DX_TAS.bat**

Example :

Master.xml	
Login.xml – Browser1	Login2.xml – Browser2
TestCases – Test-Web	TestCases- Test-Web1
Sequential	Parallel
<u>Replay logs:</u>	<u>Replay logs:</u>
3DX-TAS execution started....! [Ljava.lang.Object;@5af97850	3DX-TAS execution started....! [Ljava.lang.Object;@6181f781 [Ljava.lang.Object;@84d7023
Feb 20, 2020 3:20:17 PM org.openqa.selenium.remote.DesiredCapabili ties chrome INFO: Using `new ChromeOptions()` is	Feb 20, 2020 3:31:35 PM org.openqa.selenium.remote.DesiredCapabili ties chrome

<p>preferred to `DesiredCapabilities.chrome()`</p> <p>Feb 20, 2020 3:20:23 PM org.openqa.selenium.remote.ProtocolHandshake createSession INFO: Detected dialect: W3C</p> <p>[Sequential Execution]</p> <p>Test Case XML: testsuites\\Test-Web\\Test-Web.xml, Suite: Login, Test Case: Test-Web, Status: In Progress</p> <p>Test-Web.xml => Login => Test-Web => <Login> Processing</p> <p>Test-Web.xml => Login => Test-Web => <SwitchToDefaultContent> Processing</p> <p>Test-Web.xml => Login => Test-Web => <Logout> Processing</p> <p>Test Case XML: testsuites\\Test-Web\\Test-Web.xml, Suite: Login, Test Case: Test-Web, Status: Pass</p> <p>=====</p> <p>Login</p> <p>Total tests run: 1, Passes: 1, Failures: 0, Skips: 0</p> <p>=====</p> <p>[Ljava.lang.Object;@6ff65192</p> <p>Test Case XML: testsuites\\Test-Web1\\Test-Web1.xml, Suite: Login2, Test Case: Test-Web1, Status: In Progress</p> <p>Test-Web1.xml => Login2 => Test-Web1 => <Login> Processing</p> <p>Test-Web1.xml => Login2 => Test-Web1 => <SwitchToDefaultContent> Processing</p> <p>Test-Web1.xml => Login2 => Test-Web1 => <Logout> Processing</p> <p>Test Case XML: testsuites\\Test-Web1\\Test-Web1.xml, Suite: Login2, Test Case: Test-Web1, Status: Pass</p> <p>=====</p> <p>Login2</p> <p>Total tests run: 1, Passes: 1, Failures: 0, Skips: 0</p> <p>=====</p> <p>Master</p> <p>Total tests run: 2, Passes: 2, Failures: 0,</p>	<p>INFO: Using `new ChromeOptions()` is preferred to `DesiredCapabilities.chrome()`` Feb 20, 2020 3:31:35 PM org.openqa.selenium.remote.DesiredCapabilities chrome</p> <p>INFO: Using `new ChromeOptions()` is preferred to `DesiredCapabilities.chrome()``</p> <p>Feb 20, 2020 3:31:39 PM org.openqa.selenium.remote.ProtocolHandshake createSession INFO: Detected dialect: W3C</p> <p>Feb 20, 2020 3:31:39 PM org.openqa.selenium.remote.ProtocolHandshake createSession INFO: Detected dialect: W3C</p> <p>[Parallel Execution]</p> <p>Test Case XML: testsuites\\Test-Web1\\Test-Web1.xml, Suite: Login2, Test Case: Test-Web1, Status: In Progress</p> <p>Test-Web1.xml => Login2 => Test-Web1 => <Login> Processing</p> <p>Test-Web1.xml => Login2 => Test-Web1 => <SwitchToDefaultContent> Processing</p> <p>Test-Web1.xml => Login2 => Test-Web1 => <Logout> Processing</p> <p>Test Case XML: testsuites\\Test-Web1\\Test-Web1.xml, Suite: Login2, Test Case: Test-Web1, Status: Pass</p> <p>Test-Web1.xml => Login2 => Test-Web1 => <SwitchToDefaultContent> Processing</p> <p>=====</p> <p>Login2</p> <p>Total tests run: 1, Passes: 1, Failures: 0, Skips: 0</p> <p>=====</p> <p>Test-Web1.xml => Login2 => Test-Web1 => <Logout> Processing</p> <p>Test Case XML: testsuites\\Test-Web\\Test-Web.xml, Suite: Login, Test Case: Test-Web, Status: Pass</p> <p>=====</p> <p>Login</p> <p>Total tests run: 1, Passes: 1, Failures: 0, Skips: 0</p> <p>=====</p>
--	--

Skips: 0 ====== Don't close the application. Please wait... Generating Test Report...! Test Report Generated...! 3DX-TAS Execution completed....!	Don't close the application. Please wait... Generating Test Report...! Test Report Generated...! 3DX-TAS Execution completed....!
--	--

Adding more suite-files in suite of suites file [here it will be Master.xml] can increase browser instances but maximum limit is totally depending on property attribute we set in **3dx-tas.parallel.execution.poolsize** this is **5** in this example.

So, for above case only 5 suite files can execute in parallel, 6th suite file and onwards will start their execution once these 5 suite files finish executing test cases.

Parallel execution test cases on same machine -

3DX-TAS supports parallel execution of test cases on same machine. This approach does not need the hub and node implementation.

How to perform parallel execution on same machine -

Update suite file/test plan file path in 3DX_TAS.properties as below

3dx-tas.suitefile=Master.xml

e.g.

Master.xml file should include your suite of suites file as below

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">
<suite name="Master">
<suite-files>
<suite-file path="testsuites\\master\\Login.xml"></suite-file>
<suite-file path="testsuites\\master\\Login2.xml"></suite-file>
</suite-files>
</suite>
```

Login.xml includes

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">
<suite name="Login">
<test name="Test-Web">
<parameter name="xmlFilePath" value="testsuites\\Test-Web\\Test-Web.xml"></parameter>
<parameter name="csvFilePath" value="testsuites\\Test-Web\\Test-Web.csv"></parameter>
<classes>
<class name="com.steepgraph.ta.framework.App" />
</classes>
</test> <!-- Master -->
</suite> <!-- Suite -->
```

Login2.xml includes

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">
```

```

<suite name="Login2">
<test name="Test-Web1">
<parameter name="xmlFilePath" value="testsuites\\Test-Web1\\Test-Web1.xml"></parameter>
<parameter name="csvFilePath" value="testsuites\\Test-Web1\\Test-Web1.csv"></parameter>
<classes>
<class name="com.steepgraph.ta.framework.App" />
</classes>
</test> <!-- Master -->
</suite> <!-- Suite -->

```

Activate below properties in **3DX_TAS.properties** file. For parallel execution, use suite-of-suites file as Master suite file.

Set **3dx-tas.suitefile**=Master.xml and **3dx-tas.parallel.execution.poolsize** size as required e.g. 2

After above steps start execution of Test Suite/Test Plan using **Execute_3DX_TAS.bat**

The screenshot shows the Eclipse IDE interface. At the top, there's a code editor window titled "3DX_TAS.properties" with the following content:

```

1#-----
2# Test Execution Configuration
3#-----
4
5#Execution server name. It is only used in reporting to distinguish different server execution
63dx-tas.execution.title=QA-Run-01
7
8#Master Suite File Path.
93dx-tas.suitefile=TestPlan.xml
10
11#Name of Test Suites to execute
123dx-tas.filter.suites=
13
14#Test Cases Id to execute
153dx-tas.filter.testcases=
16
17#This property will be taken in account only if 3dx-tas.3dspace.url is blank or not present
183dx-tas.initial.url=
193dx-tas.initial.url.verify=false
20
21
22#For parallel execution of suites or test cases. set value more than 1. Default value is 1, means no parallel execution.
233dx-tas.parallel.execution.poolsize=2
24
25#This define wait interval between two xml tags in second
263dx-tas.execution.step.interval=2
27
28#selenium web driver timeout in second
293dx-tas.execution.step.timeout = 30
303dx-tas.execution.step.timeout.pollinginterval = 5
31

```

The line "3dx-tas.parallel.execution.poolsize=2" is highlighted with a red box. Below the code editor, the Eclipse interface shows the Java Application launcher with the following log output:

```

Launcher [Java Application] C:\Abhishek\eclipse-jee-2021-03-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.openjdk.hotspot\jre.full.win32.x86_64_15.0.2.v20210201-0955\jre\bin\javaw.exe (03-Sep-2021)
3DX-TAS execution started...!
[L.java.lang.Object;@5156389
[L.java.lang.Object;@4ad471f2
ERROR StatusLogger No Log4j2 configuration file found. Using default configuration (logging only errors to the console), or user programmatically.
Starting MSEdgeDriver S91t.a0r.t8i6n4g. 5M9S E(d4g4e5D6rb1vaeer3 09d1d.30.a.98a604e.e5a9e 7(749415b66b581a5e5360fdfd830a49cac09)eeae7791b651556f-3
Only local conn5e

```

The status bar at the bottom right shows the time as 01:12 PM and the date as 03-09-2021.

As seen in above image the poolsize has been set to 2, two browser instances have been launched for execution.

Once the execution is complete, one TestAutomationDebug and TestAutomationError file is generated for each of the browser instances. Check below image -

	logs
	Performance.log
	Selenium.log
	TestAutomationDebug_TestNG-tests-1.log
	TestAutomationDebug_TestNG-tests-2.log
	TestAutomationDebug.log
	TestAutomationError_TestNG-tests-1.log
	TestAutomationError_TestNG-tests-2.log
	TestAutomationError.log

Implementation of parallel test case execution on same machine is completed.

Created with the Personal Edition of HelpNDoc: [Effortlessly Convert Your Markdown Content with HelpNDoc](#)

Gracefully Shutdown

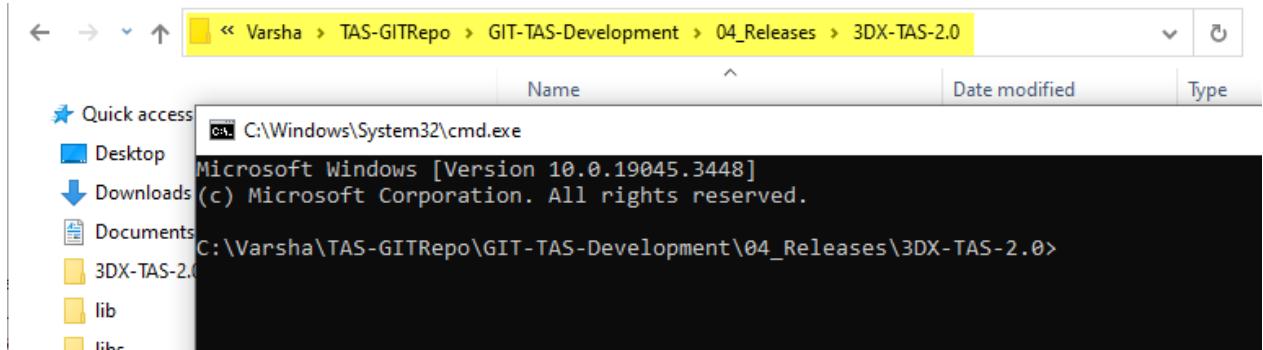
How to Gracefully shut down the Current execution

There are different approaches to shutdown current execution with gracefully

1. Go to 3DX-TAS folder, and double on **End_3DX-TAS_execution.bat (Renamed as End_SG-TAS_execution.bat)** file. By Double clicking on the file, file will be executed it will not open any terminal.

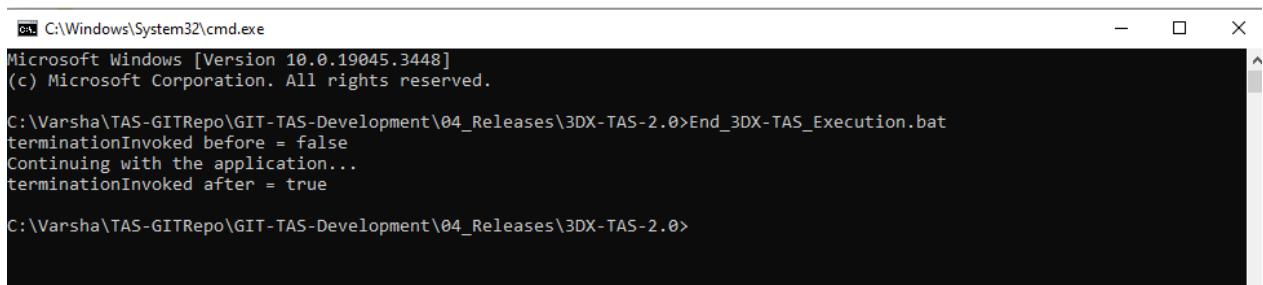
Name	Date modified	Type	Size
.settings	21-09-2023 14:38	File folder	
config	15-09-2023 16:57	File folder	
data-loader	15-09-2023 16:57	File folder	
dependencies	22-09-2023 15:43	File folder	
documents	20-09-2023 17:06	File folder	
drivers	21-09-2023 14:43	File folder	
hub	15-09-2023 16:57	File folder	
lib	25-09-2023 16:09	File folder	
license	15-09-2023 16:57	File folder	
logs	21-09-2023 14:43	File folder	
node	15-09-2023 16:57	File folder	
reports	15-09-2023 16:57	File folder	
resources	15-09-2023 16:57	File folder	
test-output	25-09-2023 17:28	File folder	
testsuites	25-09-2023 17:32	File folder	
web-recorder	15-09-2023 16:58	File folder	
.gitignore	21-09-2023 14:39	GITIGNORE File	1 KB
.project	21-09-2023 14:38	PROJECT File	1 KB
EncryptPassword	15-09-2023 16:57	Windows Batch File	1 KB
End_3DX-TAS_Execution	25-09-2023 17:02	Windows Batch File	1 KB
Execute_3DX_TAS	15-09-2023 16:57	Windows Batch File	1 KB
Execute_3DX_TAS.sh	15-09-2023 16:57	SH File	1 KB
Stop_3DX_TAS	21-09-2023 17:51	Windows Batch File	1 KB

2. To open terminal go to, 3DX-TAS folder and on the folder path right "**cmd**" and click on enter. It will open and terminal.

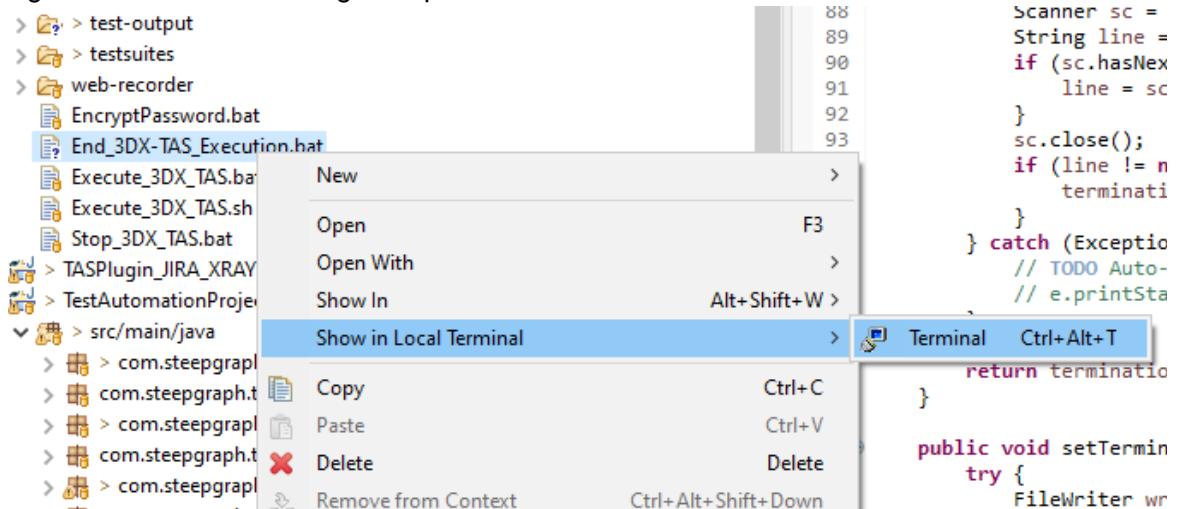


To execute the **End_3DX-TAS_execution.bat** file. Write end on command prompt click on tab button and then press enter. It will Execute the file.

3.The terminationInvoked is "**false**". When this **End_SG-TAS_execution.bat** file executed the termination will set value to "**true**". If the value is set as "**true**" it means the execution will stop. As shown in the below image.



4. User can use this file in eclipse as well. Open eclipse go to **End_3DX-TAS_execution.bat** file and right click on that file and go to option **show in Local terminal** and click on terminal.



The file will be open in terminal. To execute the file write end on terminal press on tab and then press enter. It will Execute the file. It will Execute the file same as above.

5.When the user execution first time to stop the execution gracefully for the bat file. One termination flag text file will be generated with value true.

After the execution when again you want to execute the file it will set the value false when you start the execution.

Execute_3DX_TAS.sh	15-09-2023 16:57	SH File	1 KB
Stop_3DX_TAS	21-09-2023 17:51	Windows Batch File	1 KB
terminationFlag	25-09-2023 18:18	Text Document	1 KB

If user will not delete the file then, User will not able to execute the another file after the gracefully shutdown.

6.If the user executed the **End_3DX-TAS_execution.bat** file after the completion of the testcases only the test execution stop. It not stop in the middle of any testcases. So the user can not use this file for **single test execution** scenarios. This file will only used for **multiple testcases/test plan execution with Jira integration or without Jira integration**.

7. If the user use the file in the middle of any testplan execution then it will show the result that this many test cases pass and these many test cases fail and these many test cases skip.

For example. User plan executed the test plan in, that testplan there are **5 testcases** and user want to gracefully shutdown the test execution after the second testcases. Then it will show the result as shown below.

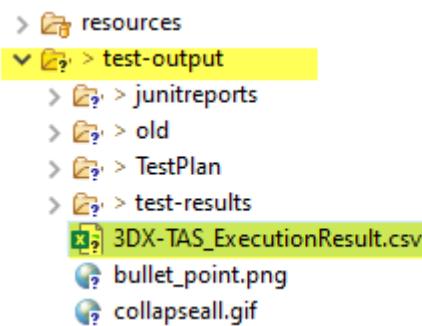
```
=====
TestPlan
Total tests run: 5, Passes: 2, Failures: 0, Skips: 3
=====

Don't close the application. Please wait...
Generating Test Report...
Test Report Generated...
3DX-TAS Execution completed....
Press any key to continue . . .
```

It will also print in terminal window that the testcases is not executed as per user input "**User terminated execution**".

```
Test Case XML: testsuites\TASNATIVE1\TAS-4\TAS-4.xml, Suite: TestPlan, Test Case: TAS-4, Status: Pass
[Ljava.lang.Object;@460510aa
User Terminated Execution
Test Case XML: testsuites\TASNATIVE1\TAS-5\TAS-5.xml, Suite: TestPlan, Test Case: TAS-5, Status: Skip
[Ljava.lang.Object;@557286ad
User Terminated Execution
Test Case XML: testsuites\TASNATIVE1\TAS-6\TAS-6.xml, Suite: TestPlan, Test Case: TAS-6, Status: Skip
[Ljava.lang.Object;@4682eba5
User Terminated Execution
Test Case XML: testsuites\TASNATIVE1\TAS-7\TAS-7.xml, Suite: TestPlan, Test Case: TAS-7, Status: Skip
```

8.If the user have terminated the execution it will show any **error message in terminal** as well as in **3DX-TAS_ExecutionResult.csv** as shown in the below image.



A	B	C	D	E	F	G	H
Suite Name	Test Case Name	Test Case Id	Execution Status	Execution Time(second)	Execution Start Time	Message	Snap Shot Name
TestPlan	TAS-2	TAS-2	Pass	19	26-09-2023 19:55		
TestPlan	TAS-5	TAS-5	Skip	0	26-09-2023 19:56	User Terminated Execution	
TestPlan	TAS-6	TAS-6	Skip	1	26-09-2023 19:56	User Terminated Execution	
TestPlan	TAS-7	TAS-7	Skip	0	26-09-2023 19:56	User Terminated Execution	
TestPlan	TAS-10	TAS-10	Skip	0	26-09-2023 19:56	User Terminated Execution	

Created with the Personal Edition of HelpNDoc: [Streamline Your Documentation Process with HelpNDoc's Project Analyzer](#)

CSV Rules

NOTE: There's certain limitation in CSV like if user want's to put single '/' as input then in CSV user need to put double '/' ('//'), as single forward slash is escaped by default in CSV.

Created with the Personal Edition of HelpNDoc: [Easily create Web Help sites](#)

Reports

Created with the Personal Edition of HelpNDoc: [Make Help Documentation a Breeze with a Help Authoring Tool](#)

Database and WebUI

Overview of 3DX-TAS Reporting module with Database:

3DX-TAS Reporting shows executed test cases in the form of reports containing tables and bar charts. This helps to analyze the complete data and statistics using graphical representation.

Home Page of 3DX-TAS Reporting:



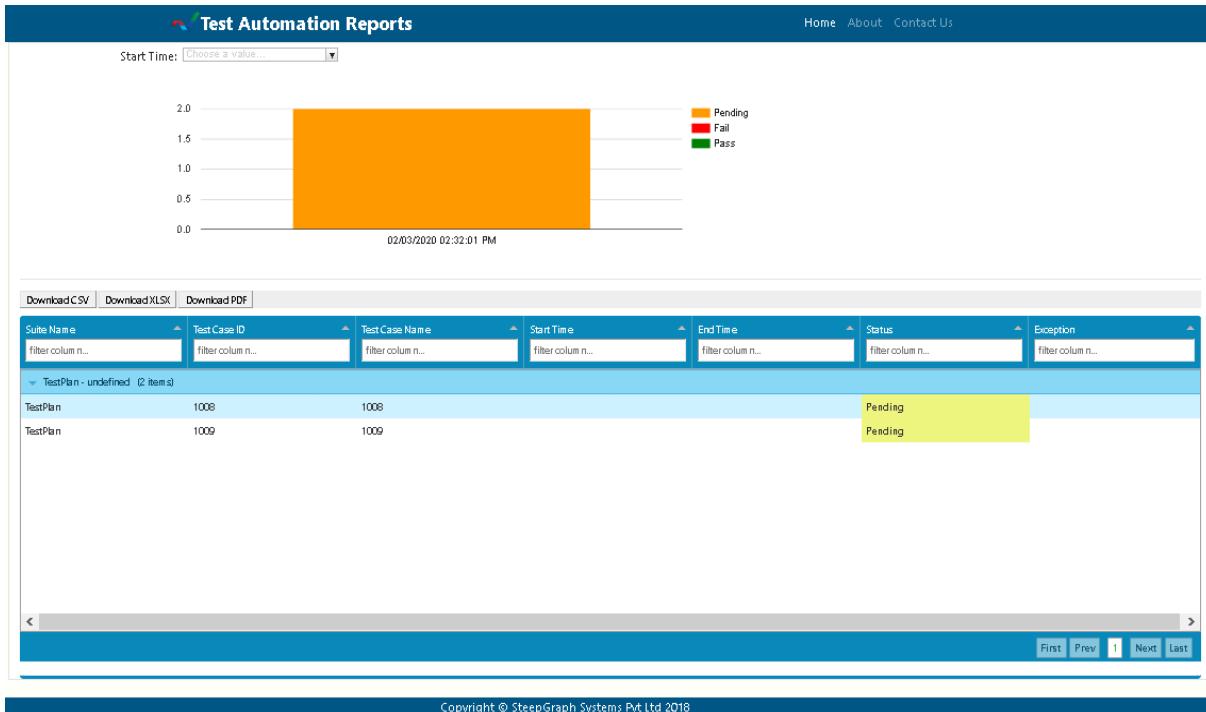
Report Gallery



Types of Reports in 3DX-TAS Reporting:

1. Latest Execution Status:

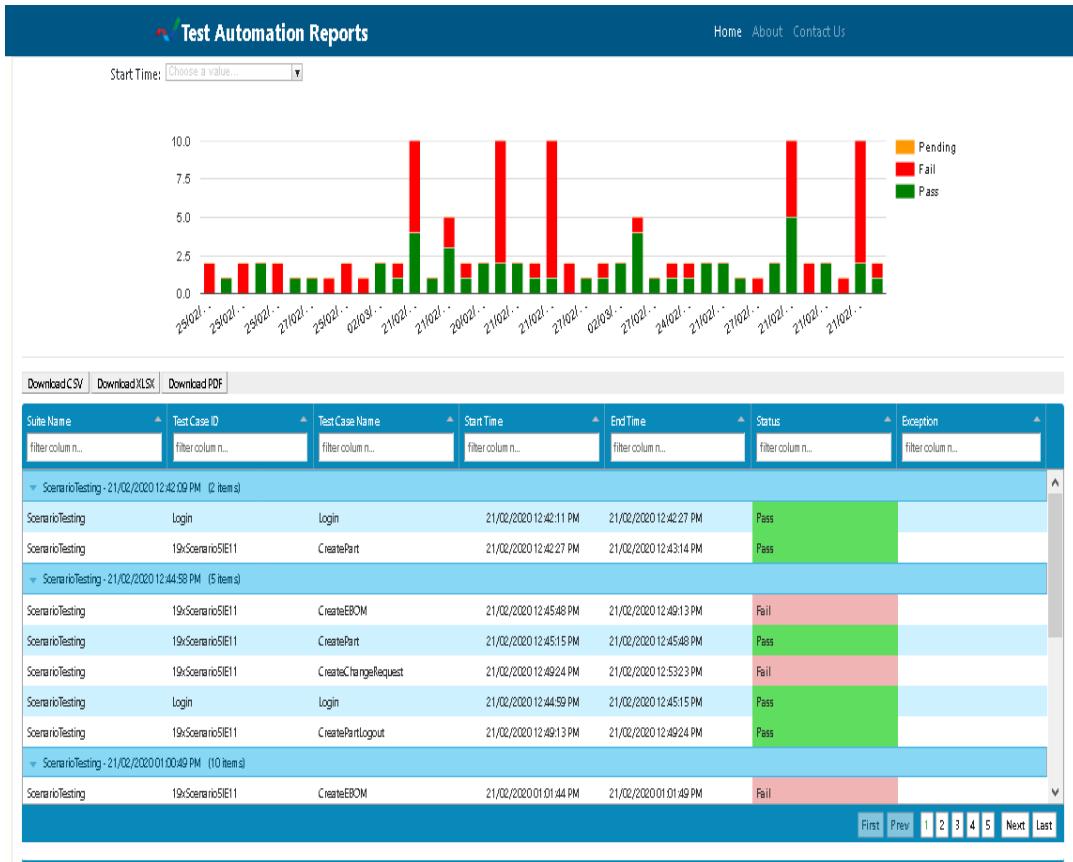
This shows the execution status of the master suite which was executed most recently and shows the test cases' suite names for that particular master suite id. This includes details like Test Case Suite Name, Test Case ID, Test Case Name, Start and End Time and Exception for each of the test cases.



2. Execution History:

This shows the test cases executed under a given particular master suite between a chosen time period.

The screenshot shows a search form titled "Form". It has three input fields: "Suite Name:" containing "TestAutomation", "Start Date:" containing "01/01/2020", and "End Date:" containing "20/04/2020". At the bottom are two buttons: a green "Submit" button and a red "Cancel" button.



3. Reports Comparison:

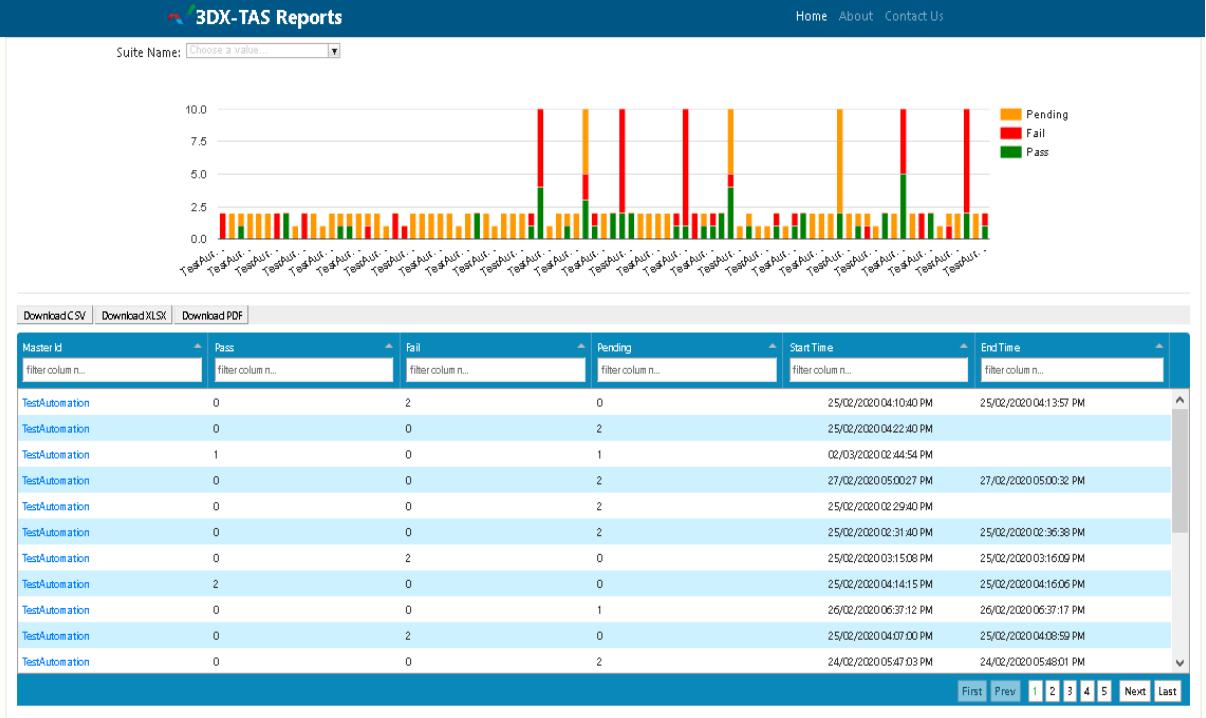
This Report shows comparison of two test case suites on the basis of suite names & dates.

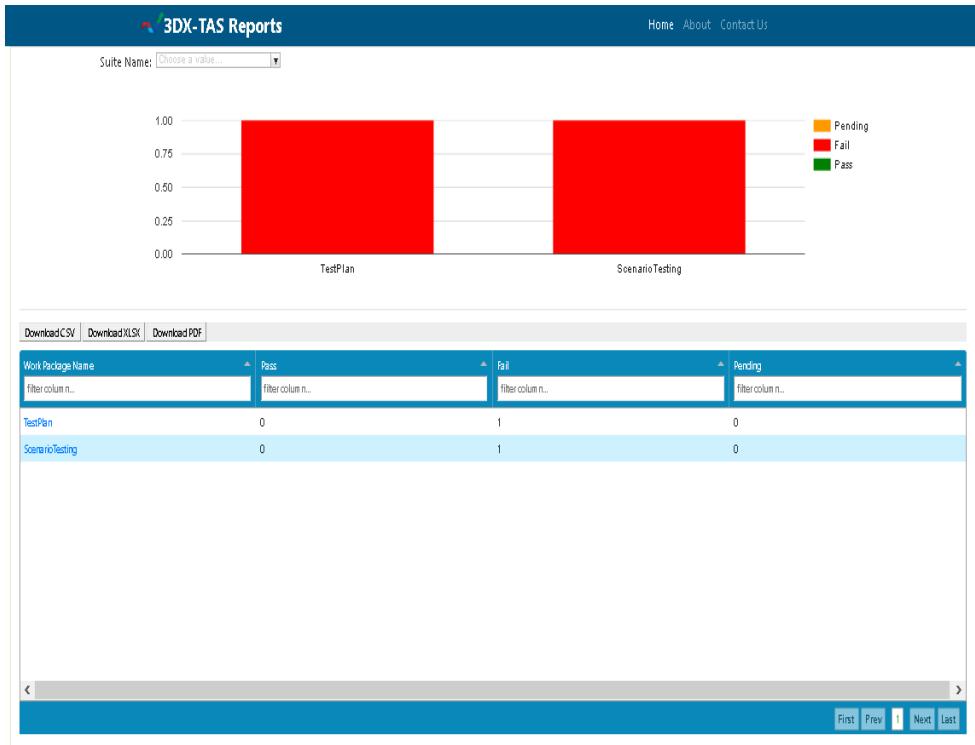
The figure shows a screenshot of the 'Test Automation Reports' application. At the top, there is a navigation bar with links for Home, About, and Contact Us. Below the navigation bar is a form titled 'Form' for comparing two test case suites. The form has two sections: 'First Suite:' and 'Second Suite:'. Each section contains a dropdown menu labeled 'Select' and a date input field labeled 'Start Date'. At the bottom of the form are two buttons: 'Submit' (green) and 'Cancel' (red).



4. Overall Execution Status:

This Report shows execution status of all test suites executed till date. The first page of the report shows the list of all master suites which have Test case under them with a link in Suite name column to second page which includes list of test cases suite names under that Master suite Id. The second page contains link in Test case suite name column to third page which shows list of all the test cases of that particular Test case suite name under the same master suite id.





The screenshot shows the 3DX-TAS Reports interface. At the top, there is a navigation bar with links for Home, About, and Contact Us. Below the navigation bar is a dropdown menu labeled "Suite Name" with the option "Choose a value...". A table displays test case details with columns for Suite Name, Test Case ID, Test Case Name, Start Time, End Time, Status, and Error Message. One row is highlighted in pink, showing the status as "Fail" and the error message as "Input XML: testsuites\...". At the bottom of the page is a navigation bar with buttons for First, Prev, 1, Next, and Last.

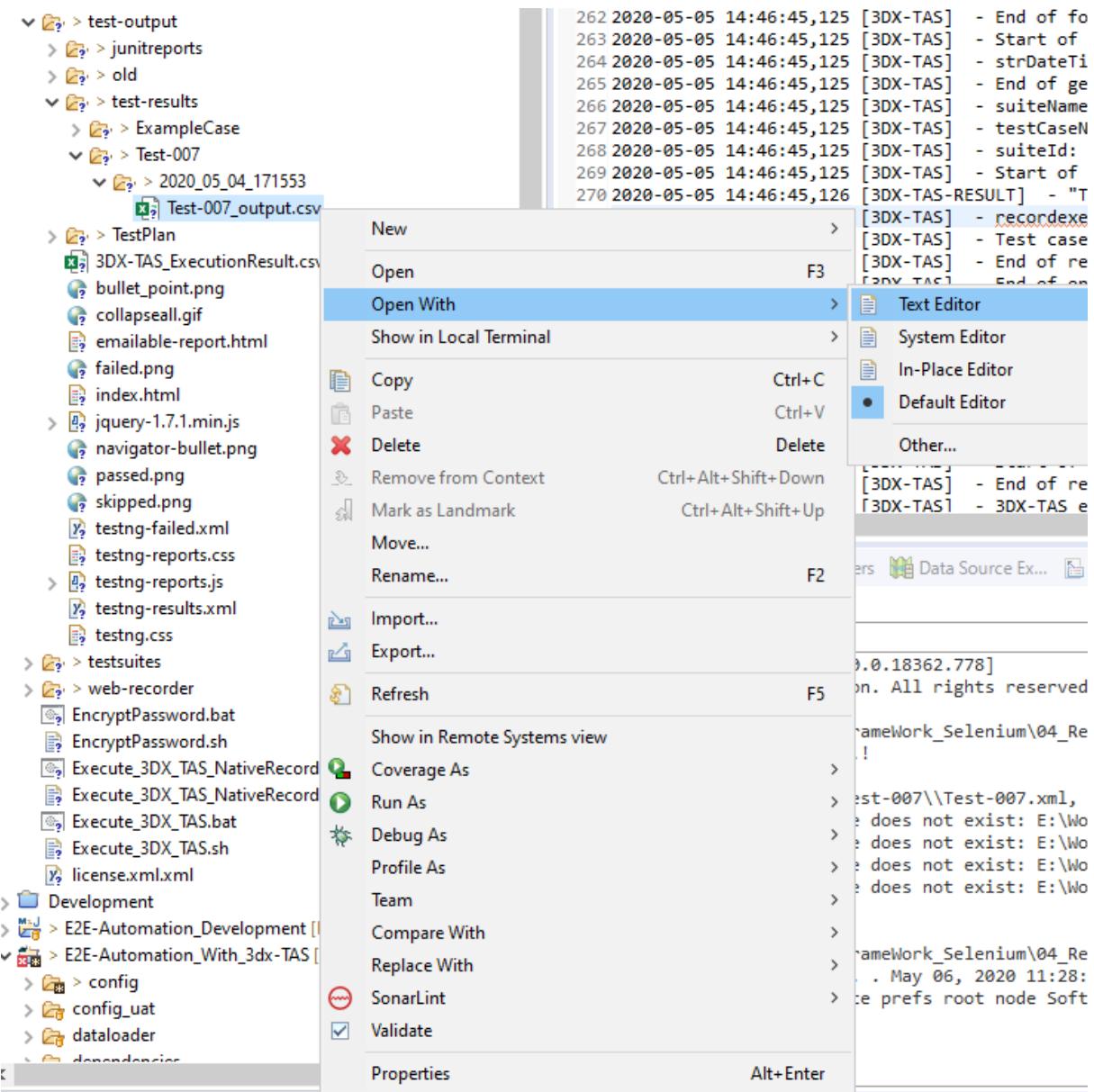
Suite Name	Test Case ID	Test Case Name	Start Time	End Time	Status	Error Message
filter column...	filter column...	filter column...	filter column...	filter column...	filter column...	filter column...
ScenarioTesting	1008	1008	25/02/2020 04:11:20 ...	25/02/2020 04:13:56 ...	Fail	Input XML: testsuites\...

Created with the Personal Edition of HelpNDoc: [Experience a User-Friendly Interface with HelpNDoc's Documentation Tool](#)

CSV

3DX-TAS Executor will generate two types of reports by default. Whenever user executes any script through **Test Plan/Test suite** in 3DS-TAS, following reports are generated in **test-output** folder as a part of 3DX-TAS Executor:

- Individual Test Case CSV Reports:** The report is generated in the **test-results** folder, which is present in **test-output** folder. The test-result folder would contain your test case folder name, which will consist filename with your testcasename_output(in .csv extension), as per timestamp. Once user has executed the script, user needs to navigate to **test-output** folder, open **test-results** folder and then he needs to open folder as per timestamp and right click on ".csv" and Open it with either Text Editor or Excel Sheet, by selecting one of program from "Open With" option as displayed in the screenshot below:



Whenever these csv report is generated in test-result output folder, you would observed that it would be generated with the test data columns in it along with 3 extra columns(i.e.

Execution Status(i.e. whether PASS or FAIL), **Execution Failure Reason**(if any in case of failure) and **Notes**(if any-This would provide the AssertPageLoadTime in case it is provided or it would warning message in case there is any).

The output report of **csv** will look like below:

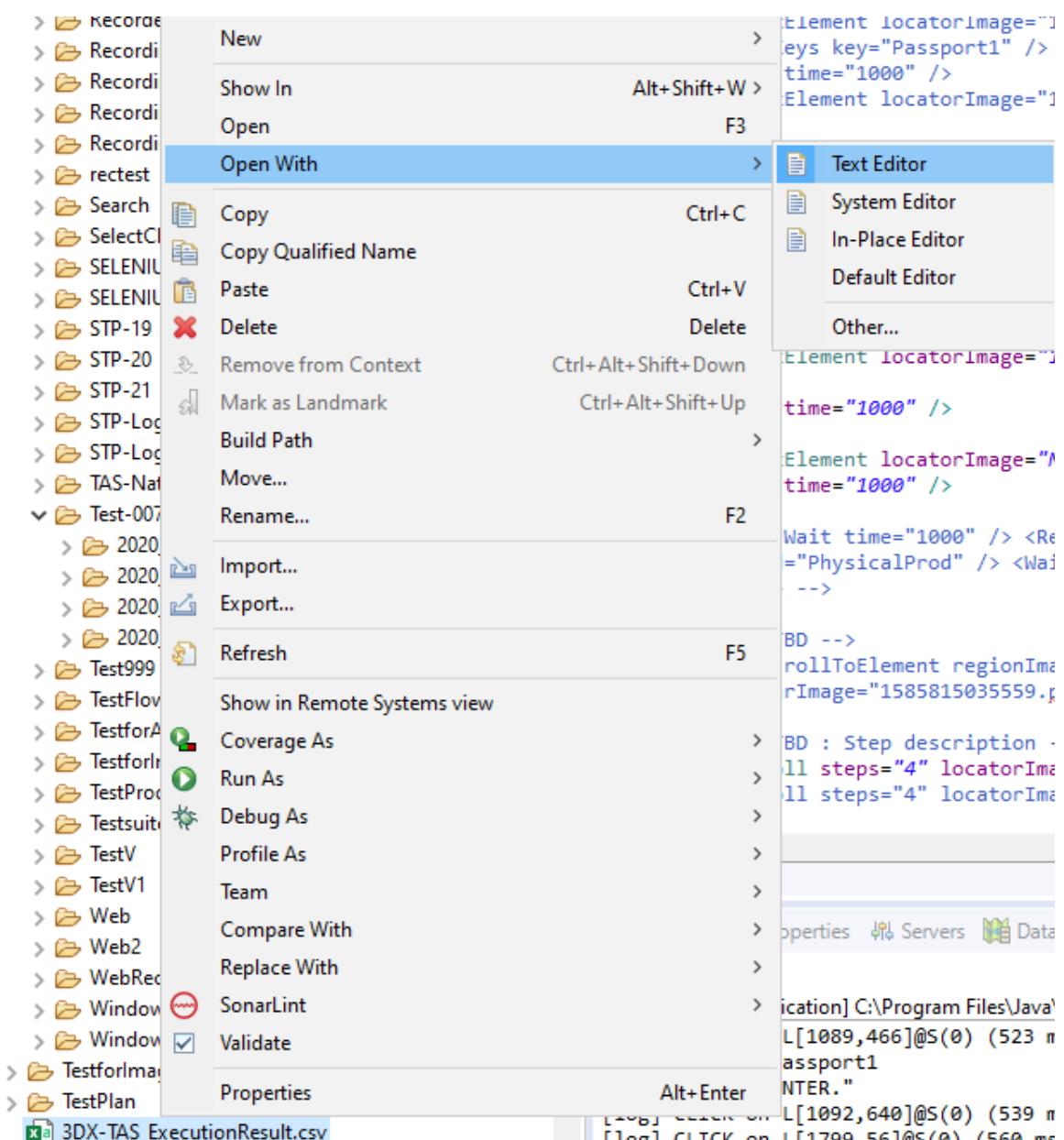
filepath	username	password	Execution Status	Execution Failure Reason(if any)	Notes(if any)
C:\Program Files\Dassault Systèmes\B421\win_b64\code\bin\CATSTART.exe	3dx_user2	T0g3aShU6v47E1iaY3!ew==	FAIL	Given image patterns are not present on screen please validate the input images : ENTER_USER_NAME.png	

NOTES:

- The reports would get stored in **./test-output/test-results** folder by default. In case

user choose to have these reports in test case folder itself, then he needs to modify **testresults.to.outputfolder=false** in **3DX_TAS.properties** file.

- Whenever user executes any test case, it will generate new report as per timestamp in test output folder, but in case user wants that every output reports get overwritten upon each execution, user needs to modify **testresults.overwrite=true**.
2. **Overall CSV Report:** The report is generated with the filename(in .csv extension) "**3DX-TAS_ExecutionResult.csv**". Once user has executed the script, user needs to navigate to **test-output** folder, right click on "**3DX-TAS_ExecutionResult.csv**" and Open it with either Text Editor or Excel Sheet, by selecting one of program from "**Open With**" option as displayed in the screenshot below:



The content/column header of .csv reports which are displayed in reports are as follows:

- Suite Name :** It displays the Suite Name which was executed. In our case we execute the scripts from TestPlan.xml. So Suite Name will appear as "TestPlan".
- Test Case Name:** It displays the Name of the Test Case's which was executed, as a part of our TestPlan.

- c) **Test Case Id:** It displays the Name of the Test Case Id's which was executed, as a part of our TestPlan.
- d) **Execution Status:** It displays the Execution Status of the Test Case's which was executed, as a part of our TestPlan.
- e) **Execution Times(second):** It displays the Execution Time(in seconds) of the Test Case's which was executed, as a part of our TestPlan.
- f) **Execution Start Time:** It displays the Execution Start Time(in Date and Time Format) of the Test Case's which was executed, as a part of our TestPlan.
- g) **Message:** It displays the Error message appeared in Test Automation/Selenium logs, in case of any kind of failure, in the Test Case's, which was executed, as a part of our TestPlan
- h) **Snap Shot Name:** It provides user the detail of the Snap Shot Name, in case of any missing Snap Shot or unidentifiable Snap Shot from our Test Script.

The output report of **csv** will look like below if the test case/script is passed:

Suite Name	Test Case Name	Test Case Id	Execution Status	Execution Time(second)	Execution Start Time	Message	Snap Shot Name
TestPlan	Test-007	Test-007	Pass	42	04-05-2020 21:37		

Suite Name	Test Case Name	Test Case Id	Execution Status	Execution Time(second)	Execution Start Time	Message	Snap Shot Name
TestPlan	Test-007	Test-007	Pass	63	05-05-2020 14:45		
TestPlan	LaunchApp-WithoutExcel	LaunchApp-WithoutExcel	Pass	1	05-05-2020 14:46		

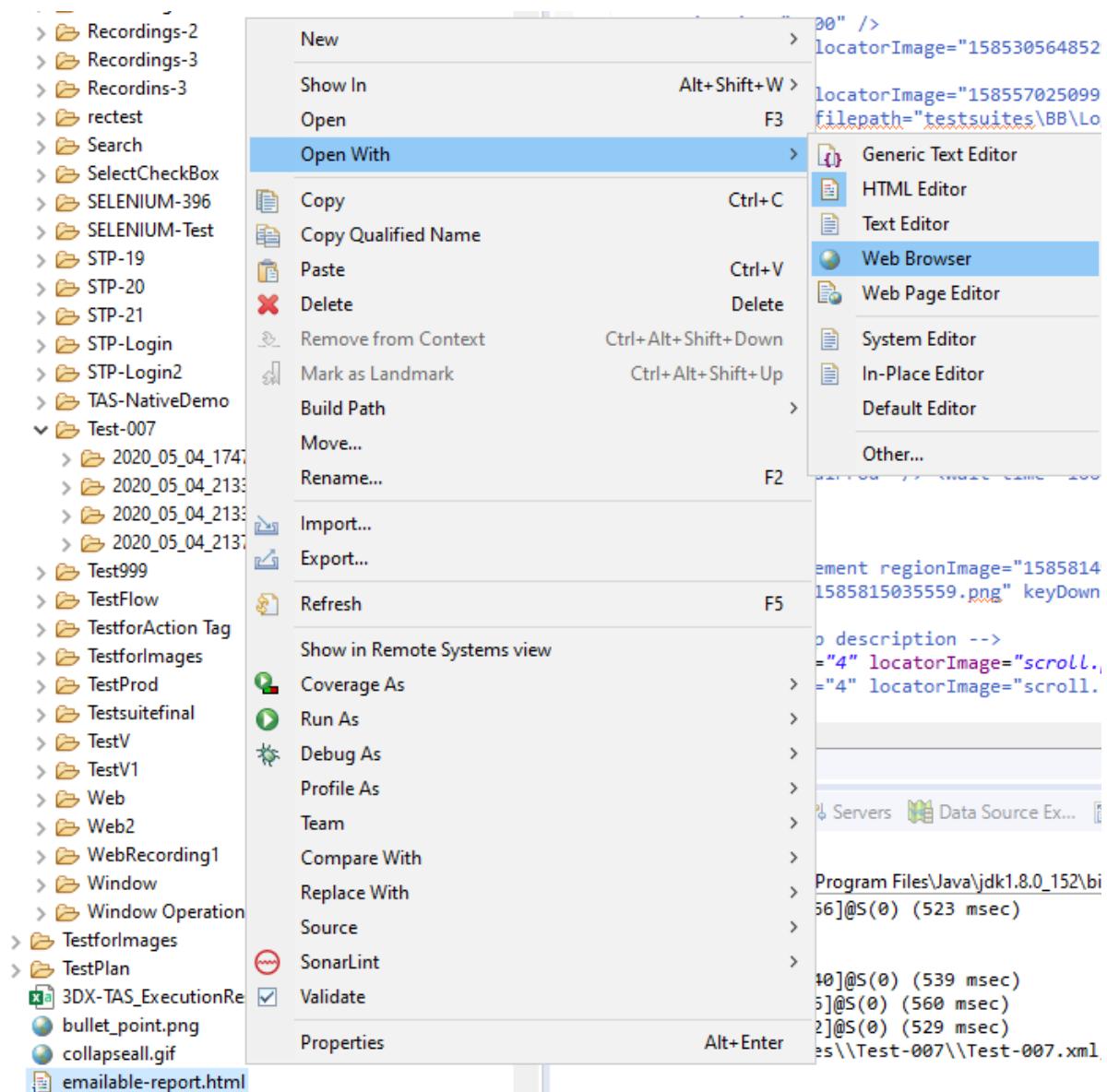
The output report of csv will look like below if the test case/script is failed:

Suite Name	Test Case Name	Test Case Id	Execution Status	Execution Time(second)	Execution Start Time	Message	Snap Shot Name
TestPlan	Test-007	Test-007	Fail	42	04-05-2020 17:47	Input XML: testsuites\ Test-007\ Test-007.xml, Suite:TestPlan, Test Case:Test-007 : Exception direction attribute is mandatory and possible values are 'up' and 'down'	

Created with the Personal Edition of HelpNDoc: [5 Reasons Why a Help Authoring Tool is Better than Microsoft Word for Documentation](#)

TestNG

- **TestNG Html emailable report:** The report is generated in **test-output** folder with the filename "**emailable-report.html**". Once user has executed the script, user needs to navigate to **test-output** folder, right click on "**emailable-report.html**" and Open it with Web Browser, as displayed in the screenshot below:



The output report of testNG will look like below if the test case/script is passed:

Test	# Passed	# Skipped	# Retried	# Failed	Time (ms)	Included Groups	Excluded Groups
TestPlan							
Test-007	1	0	0	0	44,401		
Class							
TestPlan							
Test-007 — passed							
com.steepgraph.ta.framework.App	main				1588608447554	42277	

Test-007

com.steepgraph.ta.framework.App#main

Parameter #1	Parameter #2	Parameter #3
testsuites\ Test-007\ Test-007.xml	testsuites\ Test-007\ Test-007.csv	null
Messages		
Suite: TestPlan, Test Case: Test-007, Status: Pass		

Test	# Passed	# Skipped	# Retried	# Failed	Time (ms)	Included Groups	Excluded Groups
TestPlan							
Test-007	1	0	0	0	65,078		
LaunchApp-WithoutExcel	1	0	0	0	2,801		
Total	2	0	0	0	67,879		
Class							
TestPlan							
Test-007 — passed							
com.steepgraph.ta.framework.App	main	1588670139319	62967				
LaunchApp-WithoutExcel — passed							
com.steepgraph.ta.framework.App	main	1588670204394	731				

Test-007

com.steepgraph.ta.framework.App#main

Parameter #1	Parameter #2	Parameter #3
testsuites\ Test-007\ Test-007.xml	testsuites\ Test-007\ Test-007.csv	null
Messages		
Suite: TestPlan, Test Case: Test-007, Status: Pass		

[back to summary](#)

LaunchApp-WithoutExcel

com.steepgraph.ta.framework.App#main

Parameter #1	Parameter #2	Parameter #3
testsuites\ LaunchApp-WithoutExcel\ LaunchApp-WithoutExcel.xml	testsuites\ LaunchApp-WithoutExcel\ LaunchApp-WithoutExcel.csv	null
Messages		
Suite: TestPlan, Test Case: LaunchApp-WithoutExcel, Status: Pass		

[back to summary](#)

The output report of testNG will look like below if the test case/script is failed:

The screenshot shows a TestNG report window with the following details:

- Test Plan:** TestPlan
- Test Case:** Test-007
- Status:** Failed (1 out of 1 test passed)
- Time (ms):** 62,701
- Stack Trace:**

```
com.steepgraph.ta.framework.App#main
...
java.lang.AssertionError: Input XML: testsuites\Test-007\Test-007.xml, Suite:TestPlan, Test Case:Test-007 : Exception direction attribute is mandatory and possible values are 'up' and 'down'
    at com.steepgraph.ta.framework.App.main(App.java:62)
    at java.util.ArrayList.forEach(ArrayList.java:1257)
    at com.steepgraph.ta.framework.MasterApp.start(MasterApp.java:135)
    at com.steepgraph.ta.framework.Launcher.main(Launcher.java:14)
...
Removed 25 stack frames
```

Created with the Personal Edition of HelpNDoc: [Why Microsoft Word Isn't Cut Out for Documentation: The Benefits of a Help Authoring Tool](#)

JIRA

Created with the Personal Edition of HelpNDoc: [Effortlessly upgrade your WinHelp HLP help files to CHM with HelpNDoc](#)

JIRA XRAY Integration

Important Points to Note

- JIRA is a web application that provides a private website to an individual or a set of requested users belonging to the same company/project.
- JIRA can be run as a Windows Service at the server side.
- JIRA is a pure Java based application and supports all OS platforms like Windows, Linux of different versions or MAC, etc., those satisfy JDK/JRE requirements.
- JIRA supports all famous browsers like Chrome, IE, Mozilla and Safari.
- It supports Mobile browsers as well in mobile views.
-

System Requirements

Since JIRA is a web-application, it follows the concept of client/server. It means that JIRA can be installed centrally on a server and users can interact with it through web-browsers using a website from any computer.

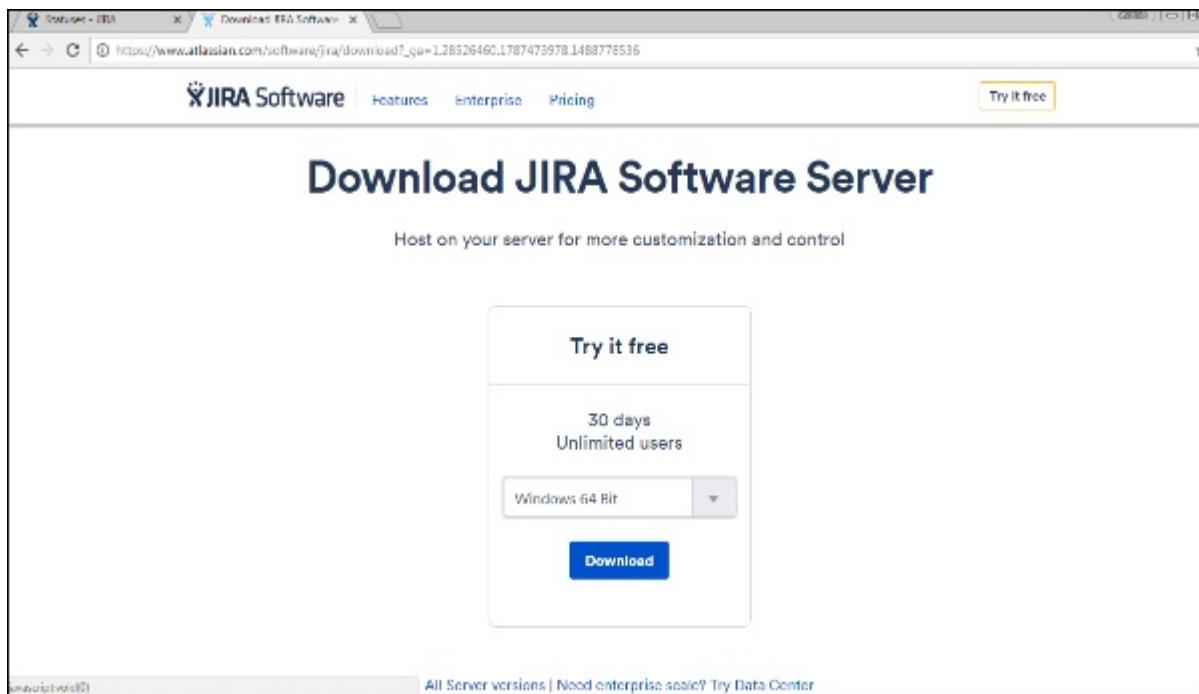
- Browser – JavaScript should be enabled, the user is recommended not to use any script-blocking tool like NoScript to access full functionality of JIRA.
- JDK/JRE – It is recommended to update JRE/JDK with the latest version. JIRA 6.4 recommends using JRE/JDK version 8.

Since our scope is to consume the JIRA application as end users, we can ignore the server side requirements.

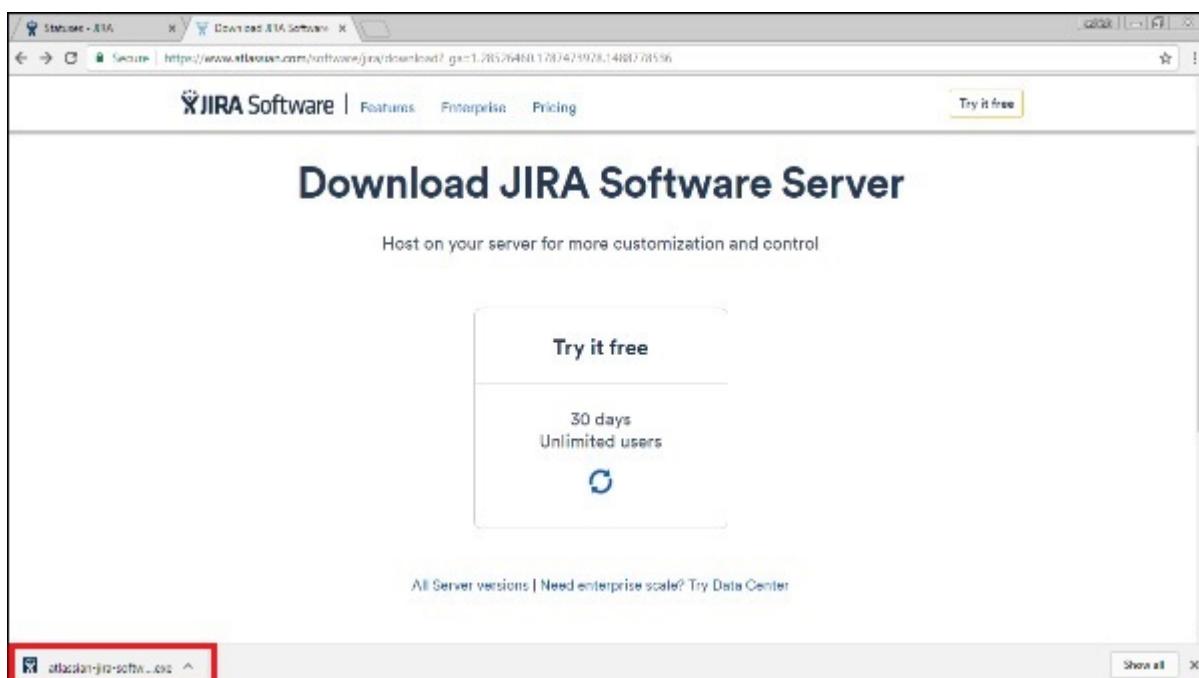
Installation at the Server Side

- JIRA follows the Client/Server concept. At the server side, JIRA must be installed before using it as end user.
- At the server side, JIRA must connect with relation database to store issues/application data.
- Download the JIRA Windows Installer.exe file from the following link – <https://www.atlassian.com/software/jira/download?ga=1.28526460.1787473978.1488778536>.
- Select the OS type and click on Download.

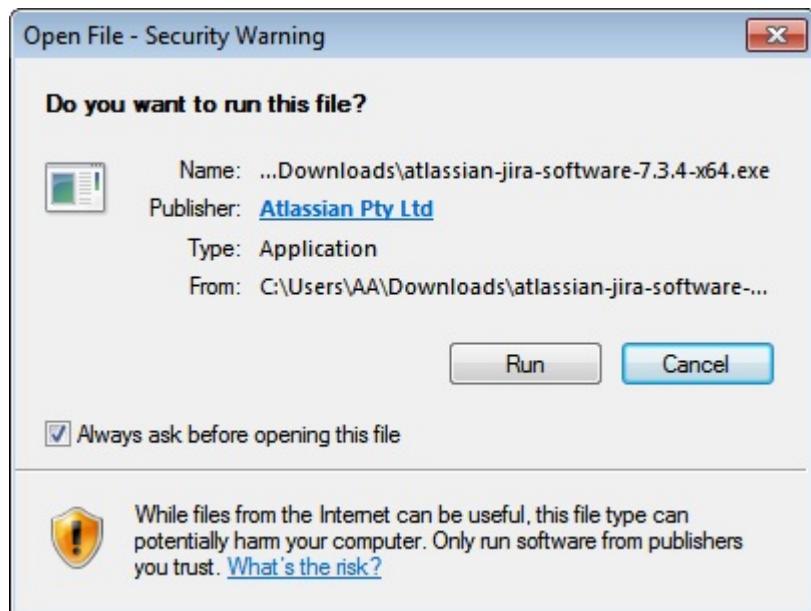
The following screenshot shows how to download the .exe file for a specific OS.



Run the .exe file to run the installation wizard. The following screenshot shows the downloaded .exe file.



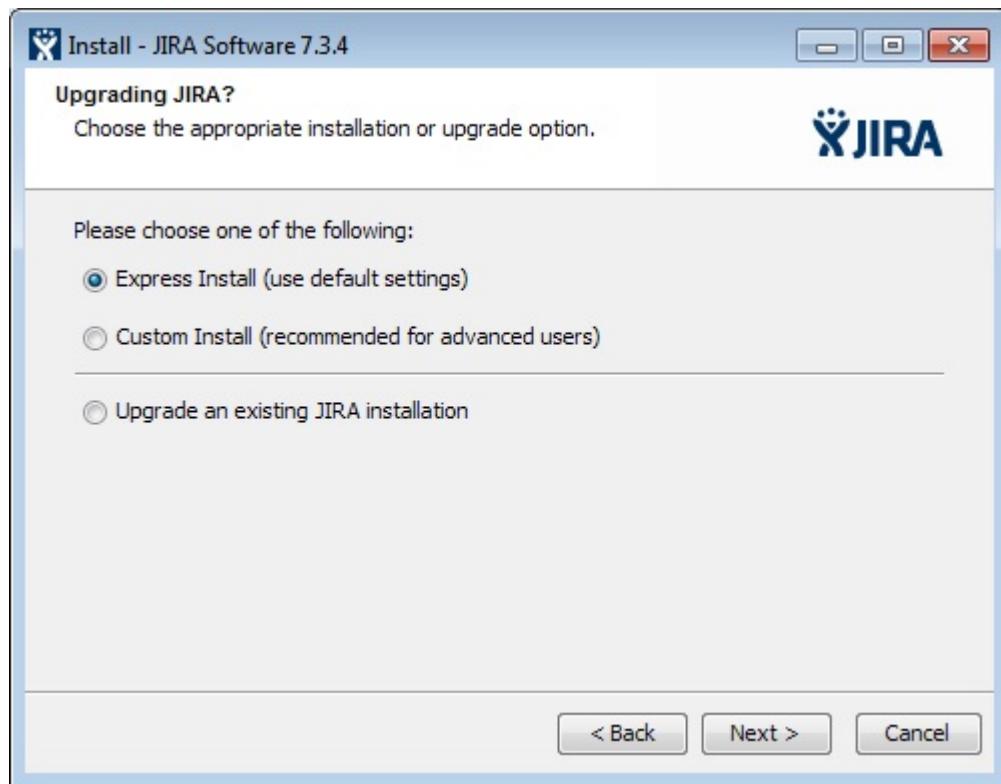
After clicking the .exe file, the Run confirmation pop-up displays, click on RUN. The following screenshot shows the RUN confirmation pop-up.



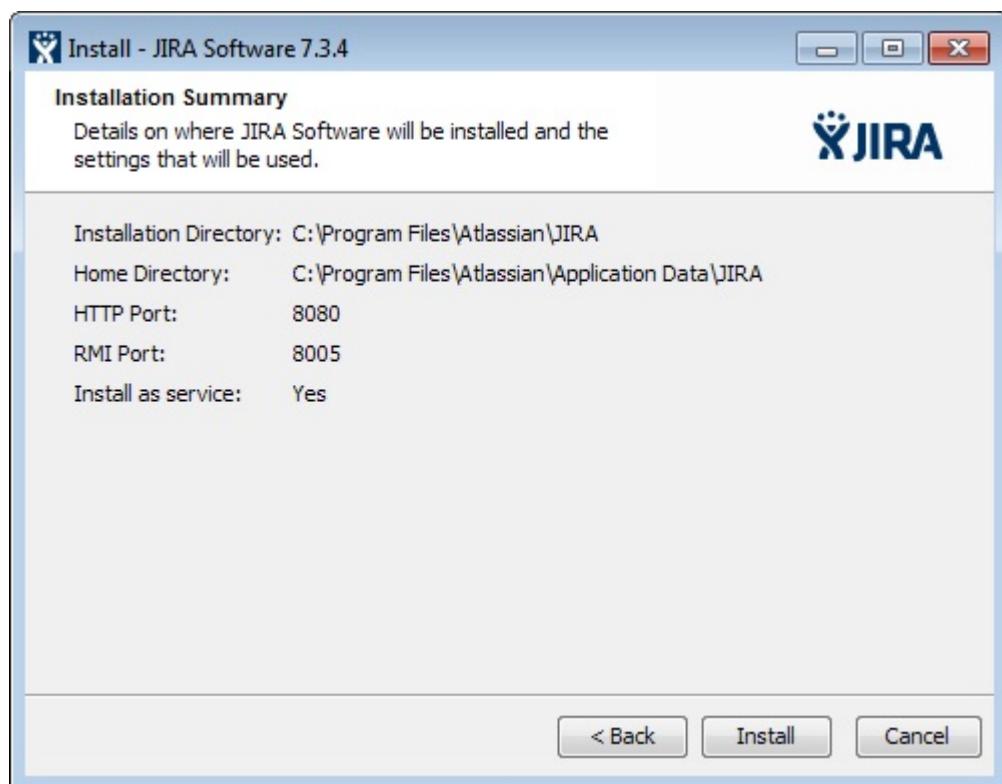
The following JIRA installation wizard displays, click on Next.



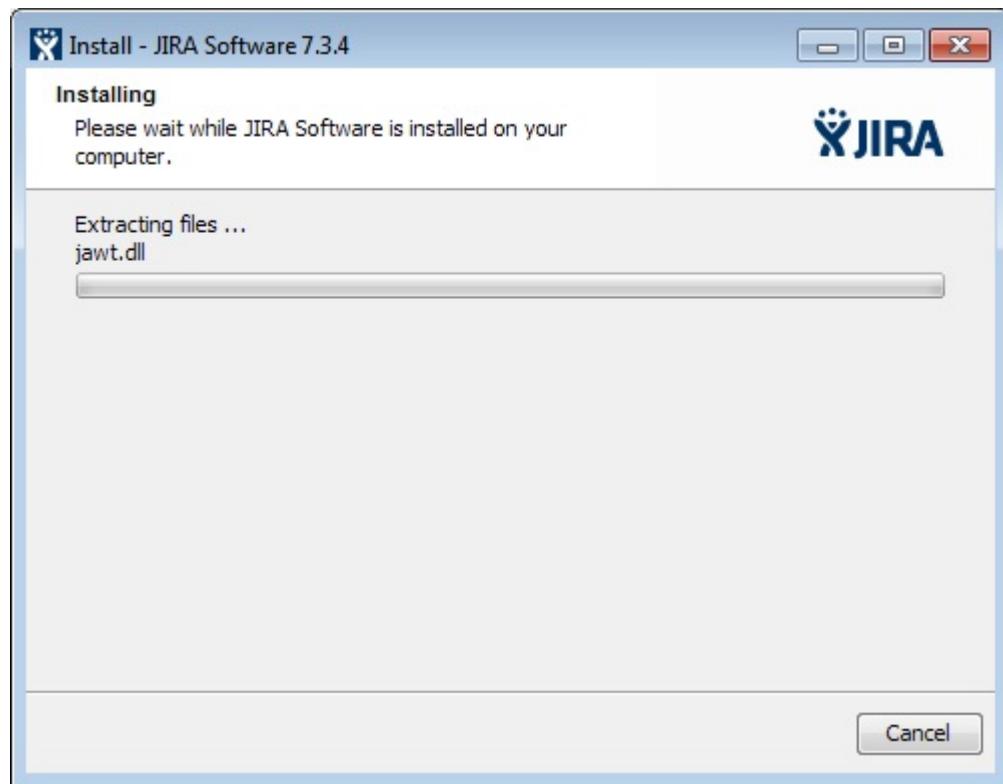
Choose the appropriate installation option as shown in following screenshot and then click on Next.



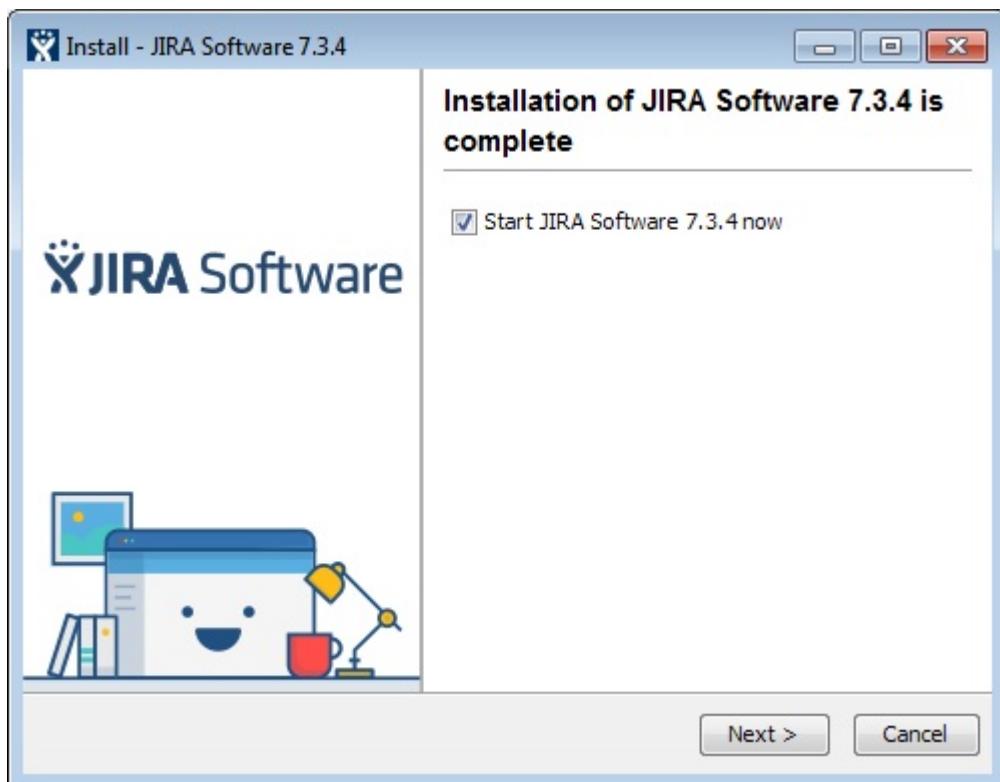
The installation summary is displayed with the Destination Directory, Home Directory, TCP Ports, etc., as shown in the following screenshot.



Click on Install. JIRA will start installing as displayed in following screenshot. It takes a couple of minutes to finish the installation.



After installation, JIRA will be started automatically if the check box to “Start JIRA Software 7.3.4 now” is checked. Then click on Next, if not, it can be accessed using the appropriate Windows Start Menu shortcut.



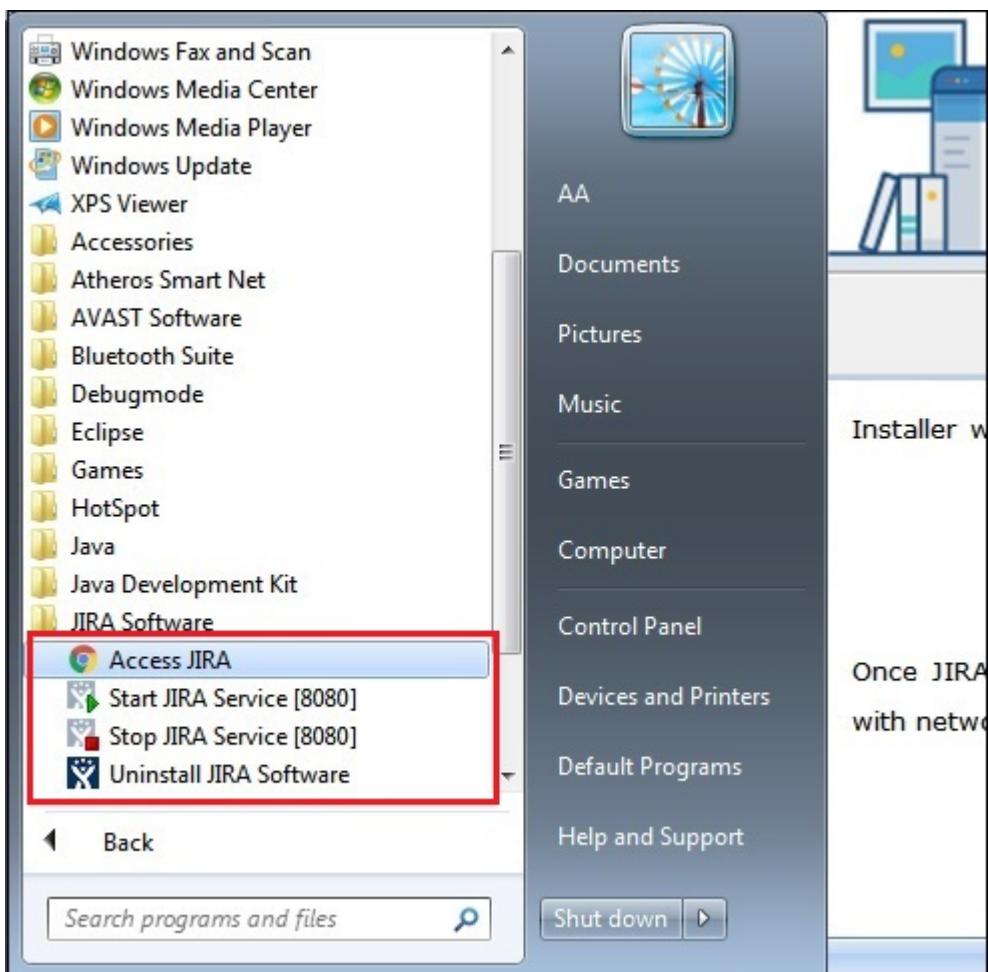
Click the Finish button.



The installer will create the following shortcuts on the start menu –

- Access JIRA
- Start JIRA Server
- Stop JIRA server
- Uninstall JIRA

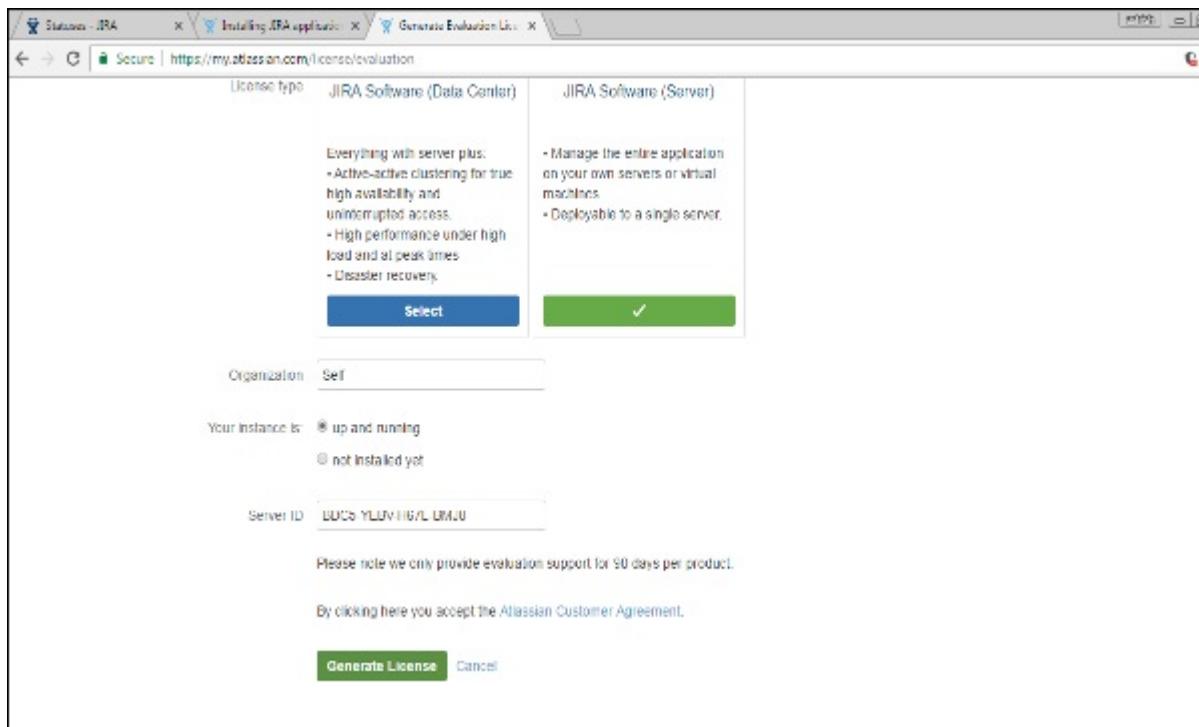
The following screenshot shows the above-mentioned shortcuts –



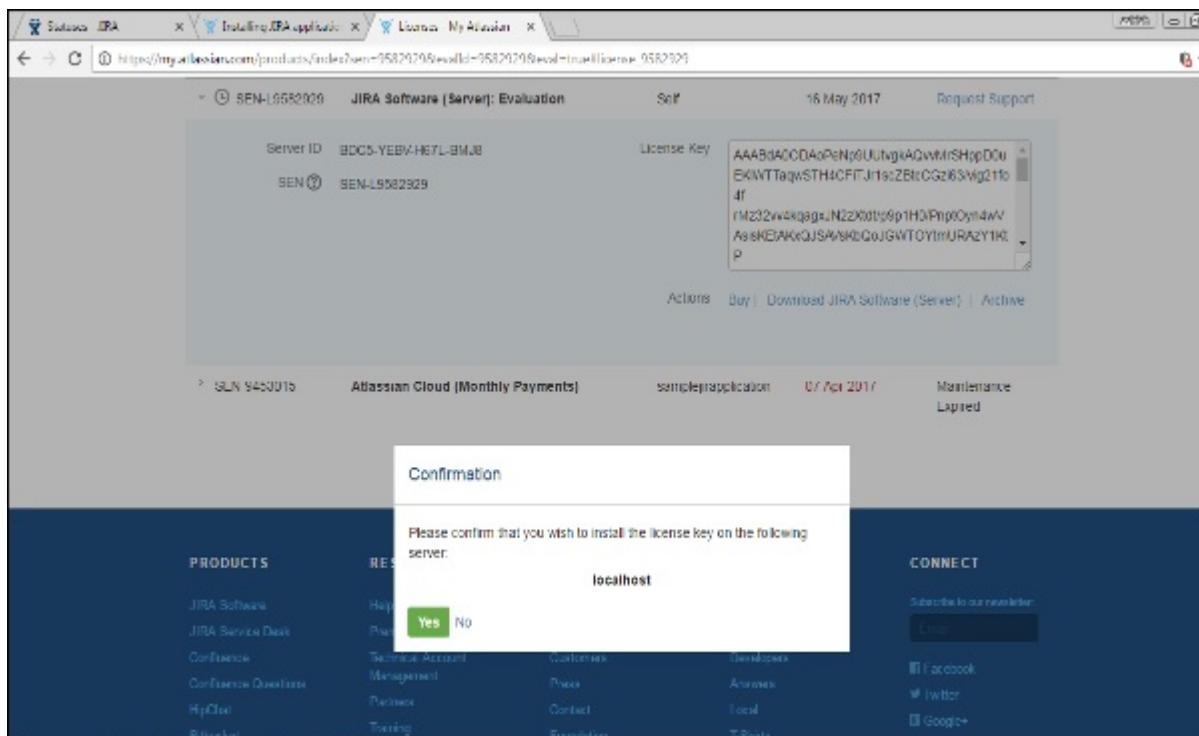
Select the license type and enter organization name as shown in following screenshot –

The screenshot shows a web browser window for the MyAtlassian website at <https://my.atlassian.com/license/evaluation>. The page title is 'New Evaluation License'. At the top, there is a message: 'Please select the proper license type.' Below this, there is a product dropdown set to 'JIRA Software'. Two license type options are listed: 'JIRA Software (Data Center)' and 'JIRA Software (Server)'. The 'JIRA Software (Data Center)' option is described as 'Everything with server plus:' with bullet points: '• Active-active clustering for true high availability and uninterrupted access.', '• High performance under high load and at peak times.', and '• Disaster recovery.'. There is a blue 'Select' button next to this option. The 'JIRA Software (Server)' option is described as 'Manage the entire application on your own servers or virtual machines.' and '• Deployable to a single server.' There is a green 'Select' button next to this option, which has a checkmark icon above it. At the bottom of the form, there is an 'Organization' field containing 'Self'.

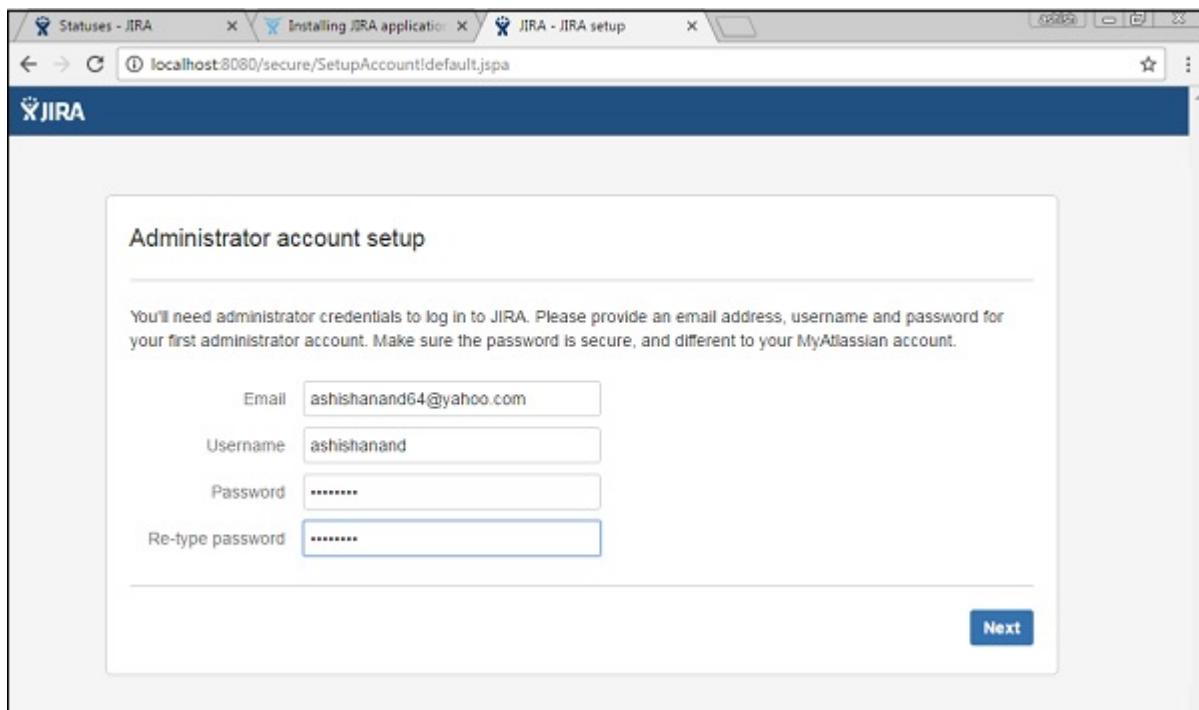
Click on Generate License



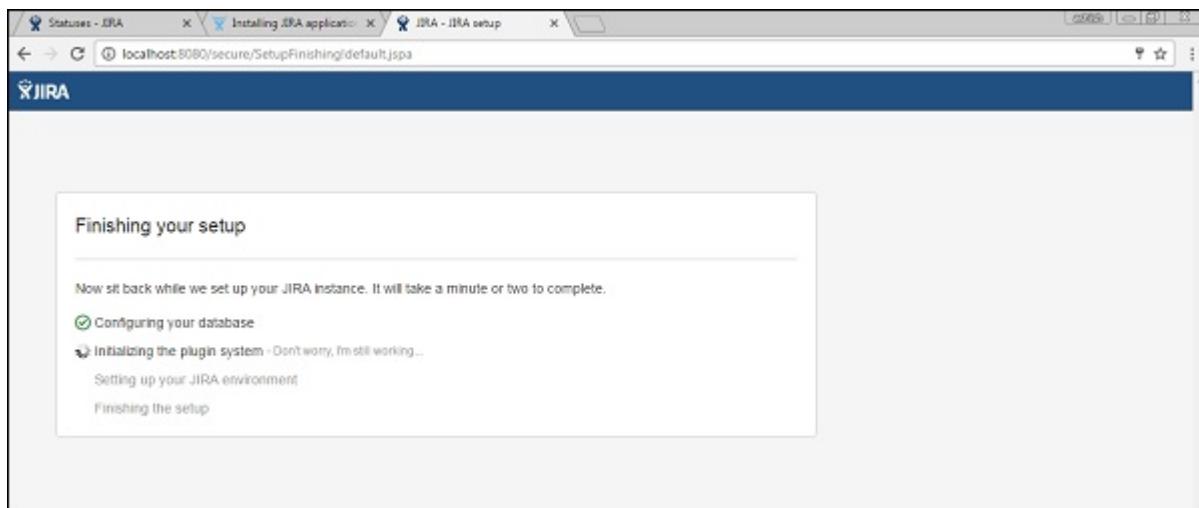
A confirmation pop-up will display as shown in the following screenshot. Click on Yes.



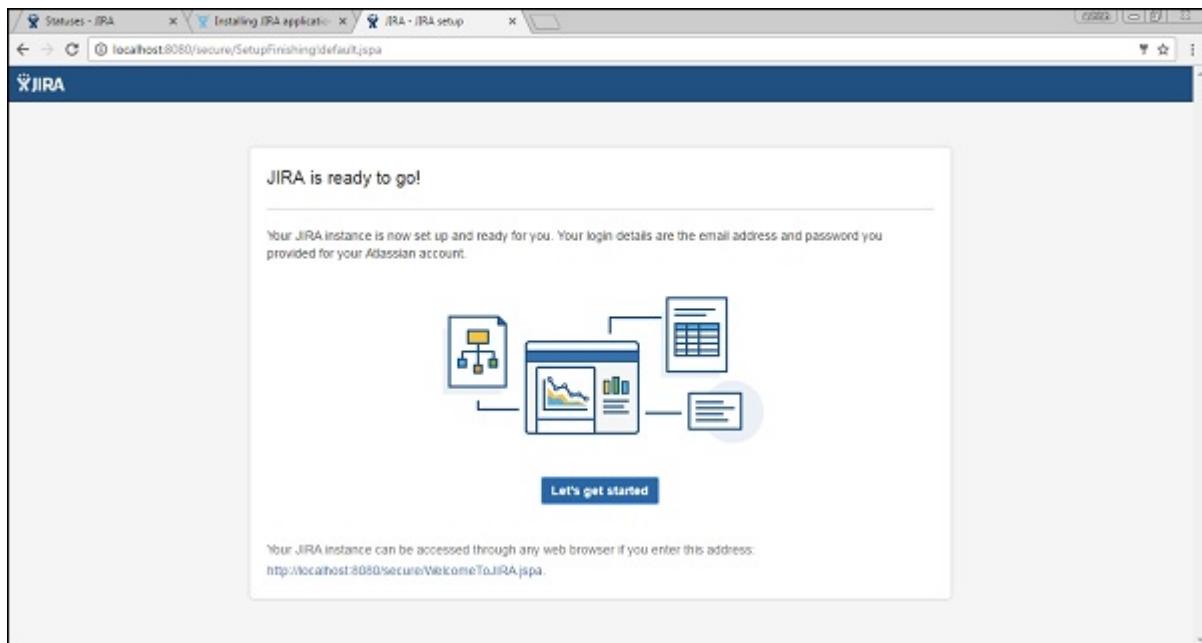
Set up the administration account as displayed in the following screenshot.



JIRA will automatically finish the setting-up as shown below –



Once JIRA finishes the setup and starts running in the server, the user can access it from a browser on any computer with network access to the JIRA server.



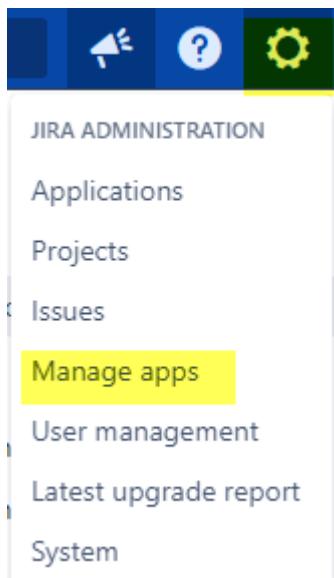
User can request to JIRA Administer to can change the limitation of integration plan as per test-cases execution.

The setting Steps are below::

1. Open JIRAXray
2. Go to setting. As soon in the below image.

The screenshot shows the Jira Software interface for the 'Native testing' issue. The left sidebar includes links for Summary, Issues, Reports, Xray Reports, Xray Test Repository, Xray Test Plan Board, Automated Steps Library, Add-ons, and Project Shortcuts. The main panel displays the issue details: Type (Test Execution), Priority (Medium), Resolution (Unresolved), Labels (None), Test Plan (TAS-1), and Test Environments (Sheetal). Buttons for Edit, Add comment, Assign, More, and Admin are visible. The right sidebar, titled 'JIRA ADMINISTRATION', contains sections for Applications, Projects, Manage apps, User management, Latest upgrade report, and System. The 'Manage apps' section is currently selected, showing options for Assignee, Reporter, Votes, Watchers, and a link to Stop watching this issue. The Dates section shows Created and Updated times as 04/Aug/23 12:23 PM.

3. Now in Setting go to the Manage Apps option.



4. Click on Miscellaneous option.

The screenshot shows the Xray settings page. On the left, under 'XRAY', there is a sidebar with the following options: 'Custom Fields', 'Issue Type Mapping', 'Requirement Coverage', 'Requirement Projects', 'Document Generator', 'Test Statuses', 'Test Step Statuses', 'Default Column Layouts', 'Parameter Values Lists', 'Integrity Checker', 'License Management', and 'Cloud Migration'. The 'Miscellaneous' option is highlighted with a yellow background. On the right, under 'Test Run Options', there are several configuration options with checkboxes:

- Set final Test Run status**: When this option is enabled, Project admins have permission to config
- Inline Execute Tests**: When this option is enabled, Test Runs can be executed inline, directly
- Restrict tests execution to assignee**: When this option is enabled, tests execution is restricted to the assignee
- Fail All Steps/Examples**: When this option is enabled, whenever the status of a Test Run is char
- Restrict automated Test status**: When this option is enabled, the status of automated Test Runs can't be
- Can Edit TestRun Dates**: When this option is enabled, the user can edit the Started On date in t
- Enable Test Run Status history log in Test Executions**: When this option is enabled, the "Updated Date" of a Test Execution is

5. After clicking on that go to option Xray Rest API option. In that option user can increase or decrease the test plan execution limit as per their requirement. default limit is set in JIRA Xray is 200 but user can exceed it above 10000 plus.

The screenshot shows the 'Xray REST API' settings page. It has a yellow header bar with the text 'Xray REST API' and 'Max results per request' followed by a text input field containing '200'. Below this, a note says 'Number of objects returned in each set of paginated REST API results.' Underneath, there are three sections: 'Associating a Test', 'Execution to a Test Plan and Importing', and 'Test results'. The 'Associating a Test' section contains the 'Add tests to test plan' checkbox and a note about associating tests with test plans. The 'Execution to a Test Plan and Importing' section contains the 'Add tests to plan' checkbox and a note about importing test results using the 'Test Plan' custom field. The 'Test results' section is partially visible.

Created with the Personal Edition of HelpNDoc: [Free EBook and documentation generator](#)

JIRA Integration

JIRA Integration with Jira Plugin:

3DX-TAS has been integrated with JIRA plugin to allow test case execution management in JIRA. Using this integration, test case executions in 3DX-TAS can be tracked in JIRA as issues. For achieving this, you can get access to the JIRA integration plugin module.

Prerequisites:

1. User should have 3DX-TAS with Jira Integration
2. JIRA Software accessible via http/https url.

The screenshot shows the 'Project settings' page in JIRA. On the left, there's a sidebar with 'Summary', 'Details', 'Re-index project', and 'Delete project'. Below that is a 'Issue types' section with 'Epic' selected. The main area has a heading 'Edit your project's workflows and screens' with a sub-instruction: 'If your workflows or screens are only used by your project, you can now edit them to make sure they suit the way your team works. Go to workflows or screens to see how they're currently set up, and to make any changes you may want. Check out our documentation for more info, or talk to your Jira administrator.' There's also a 'Actions' dropdown at the top right.

Scheme:

JIRA: Project Management Issue Type Scheme

- Bug
- Epic
- Improvement
- New Feature
- Story
- Sub-task SUB-TASK
- Task

3. Configuration:

The config folder contains the configuration property files described in the Configuration section above. Configuration for 3DX-TAS is the same if already configured. For JIRA integration to work, Jira.properties file has to be modified. Below is a screenshot of it:

```

1 #JIRA Plugin and 3DX-TAS Integration
2 -----
3
4 #Start Execution in Integration
5 jira.use.integration=true
6
7 jira.testsuites.folders = Scenarios
8
9 # Test Case/Test Plan/Story Id
10 Scenarios.jira.object.execute = JIRA-30
11
12 jira.test.environment=steepgraphdev
13
14 jira.url = http://192.168.0.225
15 jira.port = 8080
16 jira.username = test
17 jira.password = 5z0cj2DxV0N4q/+qtMuGFg==
18
19 # Generate Output
20 jira.output.File=test-output\\3DX-TAS_ExecutionResult.csv
21
22 #Update only by Admin
23 #Can be identified using url http://localhost:8080/rest/api/2/issue/<test_execution_jira_number>/transitions?expand=transitions.fields
24 jira.testExecution.promotion=21
25
26 #Upload video on the jira only if it is true.
27 #If upload is true then execution object will take lot of time to be created.
28 #This setting will work for the following TAS setting only:
29 #video.record.enable=true
30 #video.record.path=testsuites
31 #Note: For large Video Uploads to JIRA, change the JIRA setting by the Administrator.
32 # Settings --> System --> Attachments --> Edit Settings
33 # Set Attachment Size = 50MB or any desired proper value.
34 jira.video.upload=false
35
36 #Uploads .csv if it is true
37 jira.outputcsv.enable=true

```

```

#Uploads csv if it is true
jira.outputcsv.enable=true

# Jira properties
#-----

# Type of issue created after execution
# Possible values : Task, Sub-task, Story, Improvement
jira.execution.issuetype=Task

# Issue type to run test cases
# In Xray plugin : Test
jira.basic.test.issuetype=Bug

# Issue type to run test set
# In Xray plugin : Test Set
jira.basic.testset.issuetype=Improvement

# Issue type to run test plan
# In Xray plugin : Test Plan
jira.basic.testplan.issuetype>New Feature

# Issue type to run story
# In Xray plugin : Story
jira.basic.story.issuetype=Epic

# Type of Issue Link
# Possible values : Blocks, Cloners, Defect, Duplicate, Relates, Tests
jira.issueLink.type=Relates

```

Guidelines:

- Setting the property **jira.use.integration** to **true** tells 3DX-TAS to use the JIRA integration. It's default value is false, in which case 3DX-TAS executes normally without the integration.
- User could set the type of issue which would be created after execution by setting property **jira.execution.issuetype** value. The possible values for the same is Task, Sub-Task, Story and Improvement.
- Test cases to be executed must be added to a folder in the testsuites folder and it's name should be the same as specified for the property **jira.testsuites.folders**.
- In addition to the Jira.properties file, the property **3dx-tas.testresults.to.outputfolder** in 3DX_TAS.properties file should be set to **false**.
- In case user wants to capture the snapshot of the failure user needs to set **3dx-tas.snapshot.enable** key value to **true** in 3DX_TAS.properties file, else false. The snapshot would be recorded in the output folder based on key **3dx-tas.testresults.to.outputfolder**.
- If user wants to record the video of the execution, user needs to s
For every test executed, a Test Execution object/Issue is created in JIRA which contains the details of the execution and it's completion status in **Details** section.

NOTE:The current functionality covers following issue types: Task, Epic, Improvement and New Feature. This could be further extended as per project requirements.

Jira.Properties:

- **jira.use.integration:** The possible values for this key is either true or false. If user wants to start execution for Jira module, user needs to set this key value as true, else it will execute 3DX-TAS normally without taking this module into consideration.
- **jira.testsuites.folders** - In this key, user needs to mention name of folder(s) present in testsuites folder, which he/she wants to execute.
For ex: **jira.testsuites.folders=Scenarios**.

In case, user is working on multiple folders at the same time, user could provide multiple values with help of comma separator between two different folder names
 For ex. **jira.testsuites.folders**=Scenarios,TestingScenarios.

- **folder.jira.object.execute** - This key specifies the number of object which needs to be executed. It could be of type Bug/Improvement/New Feature/Epic.
 There has to be an issue in JIRA corresponding to the Bug/Improvement/New Feature/Epic with the JIRA issue's number matching the names in TAS module. The 'folder' in the key must be replaced by the name of the actual folder in the testsuites folder. There could be multiple Bug/Improvement/New Feature/Epic and multiple instances of this key as given in example below:

For ex:

Scenarios.jira.object.execute = JIRA-1 (JIRA-1 corresponds to the issue number in JIRA and the folder contains the test case files.)

TestingScenarios.jira.object.execute = JIRA-20.

NOTE:

1. The folder name which is specified in this key, is the key which is entered/used in **jira.testsuites.folders**.
 2. The type Bug/Improvement/New Feature/Epic of this jira module is similar to type Test/Test Set/Test Plan/User Story(as per testing terminologies) and it could vary and redefined as per user requirements.
- **jira.test.environment** - This key specifies the name of environment, on which test cases would be executed. This could be any string. Example:.steepgraphdev, plmint01 etc.
 - **jira.url** - This key specifies JIRA instance URL to be used for execution. et **3dx-tas.video.recording.enable** key value to **true** in 3DX_TAS.properties file, else false. The video would be recorded in the output folder based on key **3dx-tas.testresults.to.outputfolder**.
 - In 3DX_TAS.properties file, the value of **3dx-tas.testresults.overwrite** should be set to true, in order to ensure that outputs are overwritten. Also image and video are attached to newly created issue.

E.g.: **jira.url**=http://192.168.0.225.

- **jira.port**- This key defines JIRA server's port number.
 E.g.: **jira.port** = 8080
- **jira.username** - This key defines user name, which is used to connect to Jira while testing.
 E.g.: **jira.username** = test
- **jira.password** - This key specifies the password for the username defined in the key **jira.username**. The password should be entered in encrypted form.
 E.g.: **jira.password** = 5z0Cj2DxV0N4q/+qtMuGFg==
- **jira.output.File** - With the help of this key, user could keep execution results in a predefined csv file.
 E.g.: **jira.output.File**=test-output\\3DX-TAS_ExecutionResult.csv.
- **jira.testExecution.promotion** - This key defines the id of the workflow transition state to which the test execution object created for a execution, should be promoted post execution irrespective of it's completion status. It is obtained by using the jira rest hook:
http://localhost:8080/rest/api/2/issue/<test_execution_jira_number>/transitions?expand=transitions.fields where <test_execution_jira_number> is the Jira object's number.
 For ex. The url: <http://localhost:8080/rest/api/2/issue/TEST-50/transitions?expand=transitions.fields> gives the output in JSON format as below. If a user wants the test execution object to be promoted to Done state after execution, the id from the below json output for Done state is 41. This should be used in the key as :
jira.testExecution.promotion=41.
- **jira.video.upload** - This key uploads video on the jira if and only if it is set to true. The default value is false. Also, user needs to set **3dx-tas.video.recording.enable** to true in

- 3DX_TAS.properties file, if he/she wants to capture video of test execution, else false.
- **jira.outputcsv.enable** - This key is defined for uploading csv file. If user wants to upload csv file, he/she needs to set **jira.outputcsv.enable=true**, else false.
 - **jira.execution.issuetype** - This key specifies the type of issue which would be created after required execution. The possible values for this key is Task, Sub-Task, Story and Improvement.
E.g.: **jira.execution.issuetype=Task**.
 - **jira.basic.test.issuetype**- This key specifies issuetype to run test cases.
E.g.: **jira.basic.test.issuetype = Bug**
NOTE: The value of the above key could be customized easily by user as per their requirements. User could give the above key value as Improvement, New Feature etc. as per their requirements,
 - **jira.basic.testset.issuetype**- This key specifies issuetype to run test set.
E.g.: **jira.basic.testset.issuetype = Improvement**
NOTE: The value of the above key could be customized easily by user as per their requirements. User could give the above key value as Improvement, New Feature etc. as per their requirements,
 - **jira.basic.testplan.issuetype**- This key specifies issuetype to run test plan.
E.g.: **jira.basic.testplan.issuetype = New Feature**
NOTE: The value of the above key could be customized easily by user as per their requirements. User could give the above key value as Improvement, New Feature etc. as per their requirements,
 - **jira.basic.story.issuetype**- This key specifies issuetype to run test story.
E.g.: **jira.basic.testplan.issuetype = Epic**
NOTE: The value of the above key could be customized easily by user as per their requirements. User could give the above key value as Improvement, New Feature etc. as per their requirements,
 - **jira.issueLink.type**- This key specified the type of Issue Link which exists between issuetype and execution object. The possible values for this key are Blocks, Clones, Defect, Duplicate, Relates, Tests.
E.g.: **jira.issueLink.type = Relates**.

Created with the Personal Edition of HelpNDoc: [Maximize Your Productivity with HelpNDoc's Efficient User Interface](#)

JIRA Folder Structure

Executing the Test plan:

Execution can be started by running the **Execute_3DX_TAS.bat** directly or as administrator from command prompt.

encryptPassword.bat	13-04-2020 17:30	Windows Batch File	1 KB
Execute_3DX_TAS.bat	13-04-2020 17:30	Windows Batch File	1 KB
Execute_3DX_TAS.sh	13-04-2020 17:30	Shell Script	1 KB

```

Microsoft Windows [Version 10.0.19041.329]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\SGSPC042>E:

E:\>cd E:\Work\P510\04_Releases\3DX-TAS-2.0-WebOnly\3DX-TAS-2.0

E:\Work\P510\04_Releases\3DX-TAS-2.0-WebOnly\3DX-TAS-2.0>Execute_3DX_TAS.bat

```

General checks before executing Test Plan with JIRA Integration:

1. The folder specified by jira.testsuites.folders key should be present in testsuites folder in

3DX-TAS. There can be multiple folders in testsuites for testing. Accordingly the property values could be comma separated.

Example:

jira.testsuites.folders=Folder1,Folder2

2. In the property folder.jira.object.execute, folder must be replaced by the appropriate folder name from the testsuites folder.

Example:

Folder1.jira.object.execute

3. The value of the key folder.jira.object.execute should match the issue ID of the JIRA Bug/Improvement/New Feature/EPIC to be executed.

Example:

Folder1.jira.object.execute = JIRA-1, where JIRA-1 is the JIRA issue ID

4. There could be multiple instances of the key folder.jira.object.execute in case of multiple folders.

Example:

If there are two folders, folder1 and folder 2

Folder1.jira.object.execute = JIRA-1

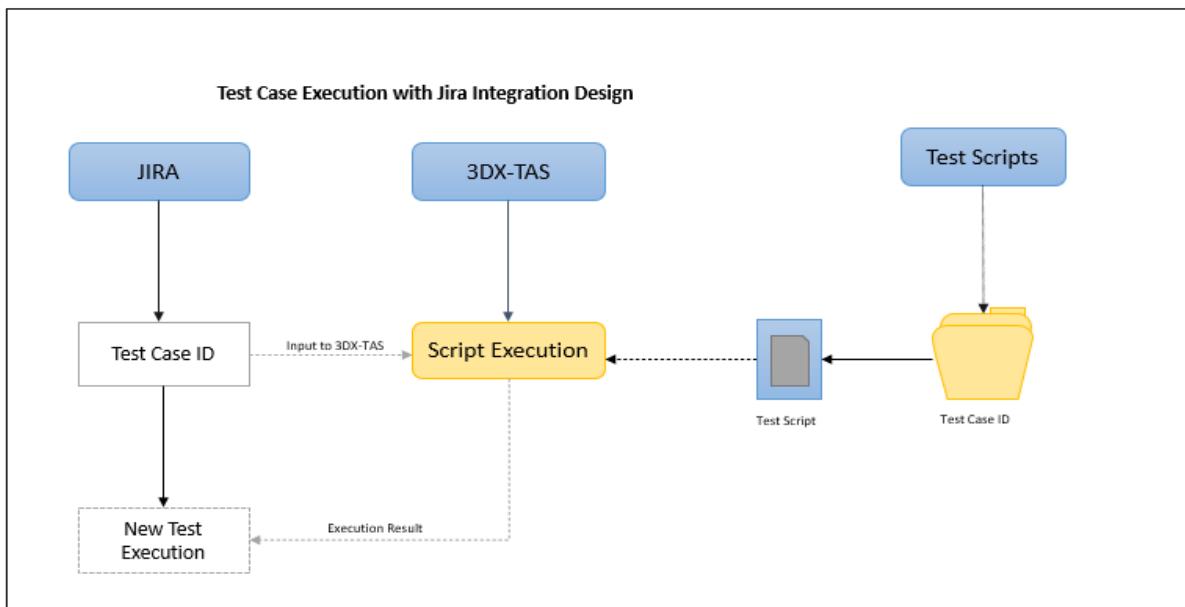
Folder2.jira.object.execute = JIRA-20

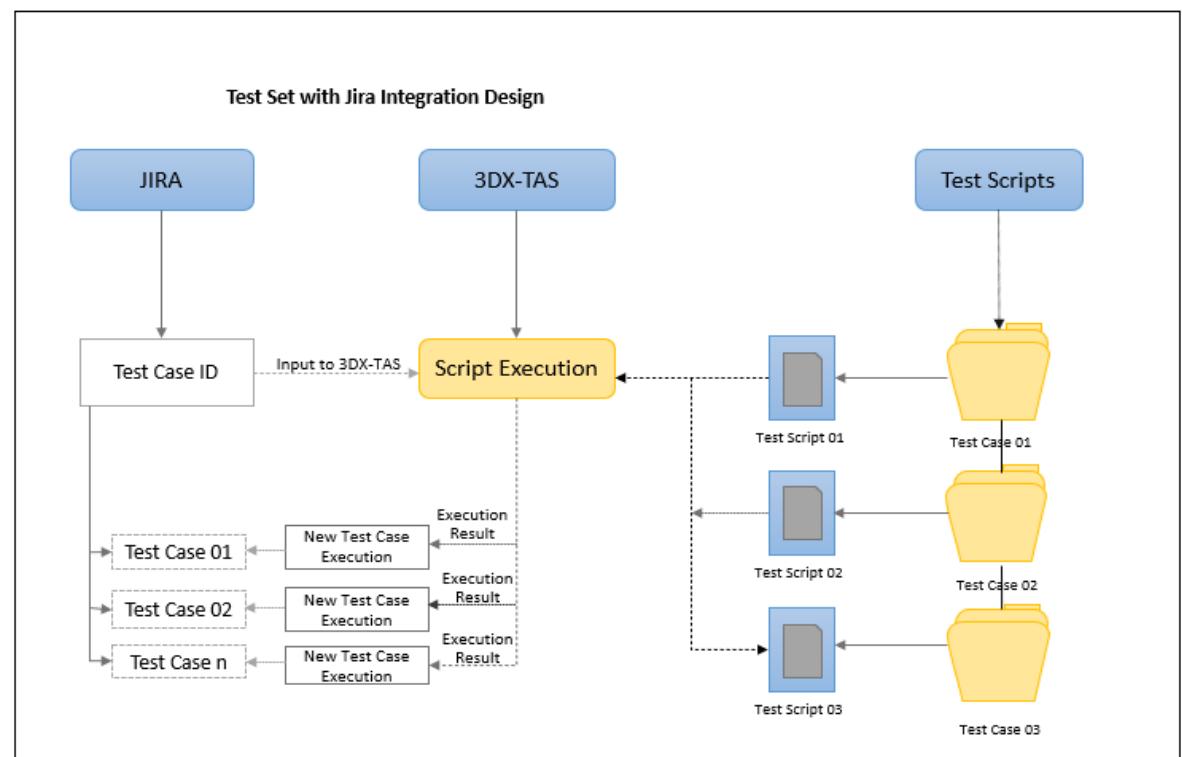
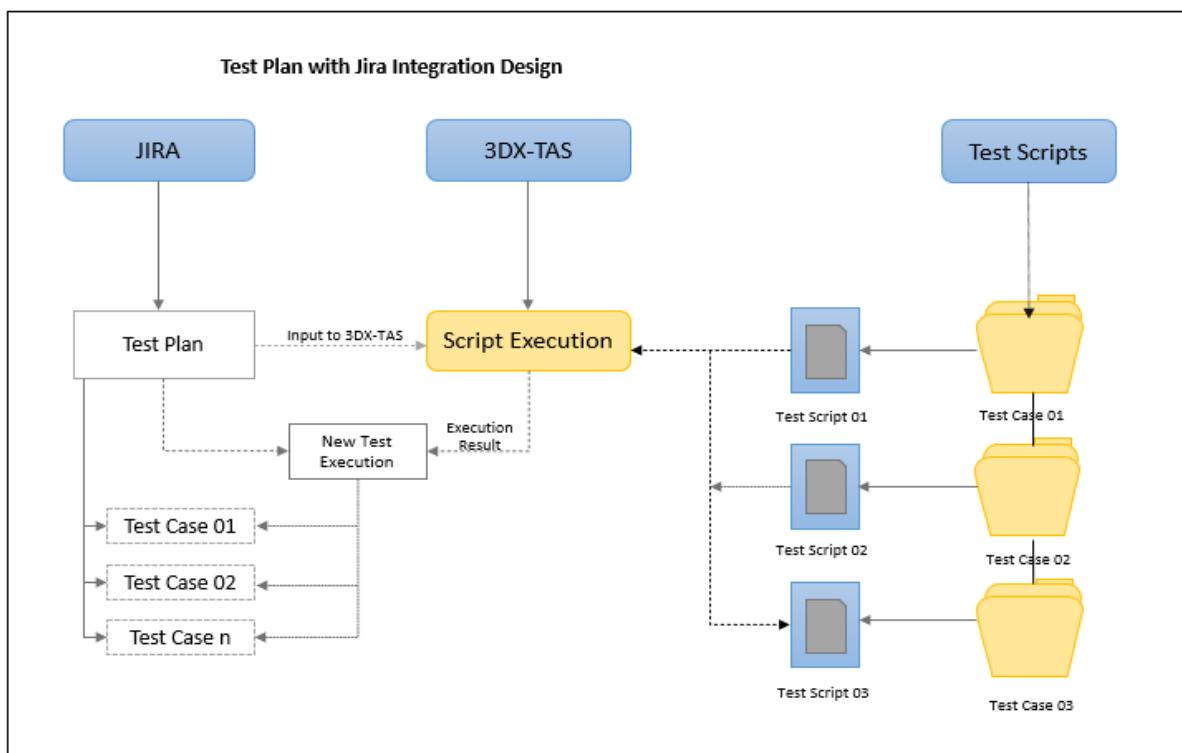
JIRA Integration Test Scenarios:

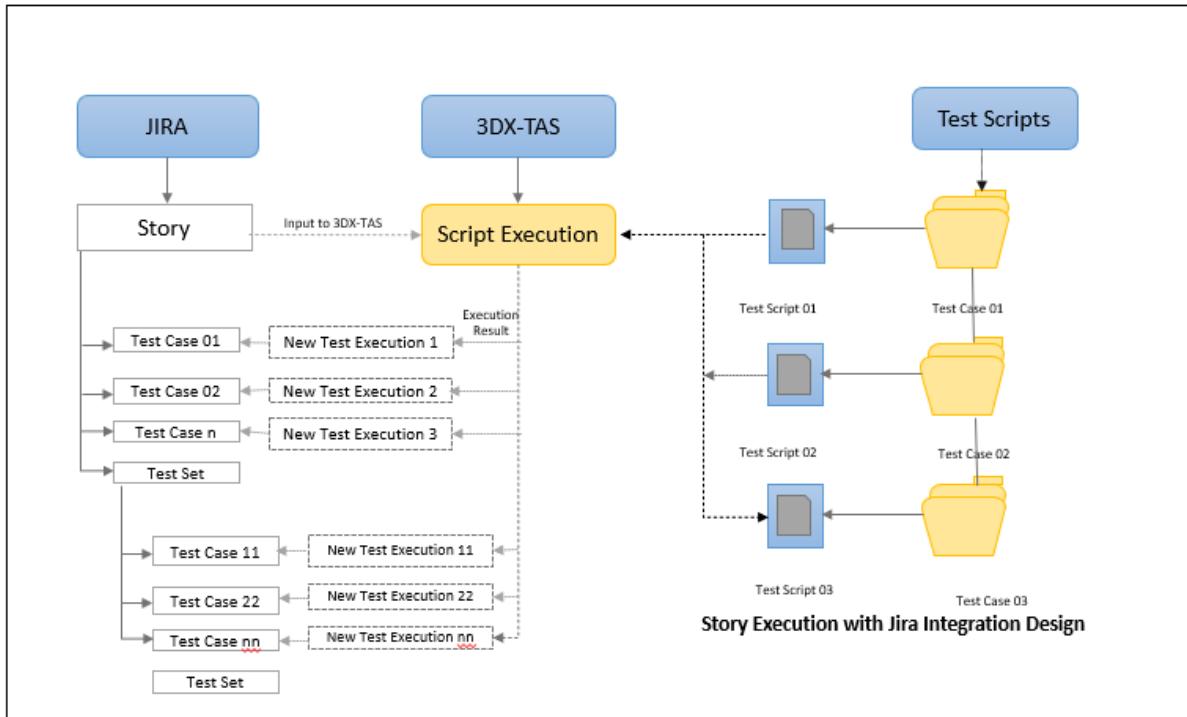
There are multiple scenarios in which the Jira integration could be used.

- Executing one or more test cases.
 - Executing test plan with one or more test cases.
 - Executing Story with test cases and/or test set containing test cases.
 - Executing a test set with test cases.

Jira Integration flow scenarios:







Examples:

1. For explaining the execution scenarios let us consider a project "TestProject" in JIRA. It has been configured to allow Jira coverage and it's issue types also include Jira issue types.
2. In TAS local setup, testsuites folder contains a folder named Scenarios in which all the relevant test cases will be present.
3. The key jira.testsuites.folders has been set to jira.testsuites.folders=Scenarios in Jira.properties file.
4. folder.jira.object.execute key will be set to appropriate values based on JIRA issue ID(s) being tested.
5. On completion of test, the Test Execution object created for the Bug will be set to Done state specified by the property jira.testExecution.promotion=21. The process to set the state is described in the Configuration section for 3DX-TAS.

1. Executing one or more test cases:

Here Bug executes functionality of Test as per values in Jira.properties.

Configuration and execution:

The JIRA issue type for execution is Test. If more than one test case is to be executed, corresponding tickets must exist in the JIRA project.

Test Login functionality

Type: Bug Priority: Medium Labels: None Status: TO DO (View Workflow) Resolution: Unresolved

Description: Click to add description

Jira.properties:

For multiple Tests, comma separated values can be given for the property TASTestingProject.jira.object.execute.

Example:

```
# Test Case/Test Plan/Story Id
Scenarios.jira.object.execute = JIRA-1,JIRA-20
```

In TestJiraProject folder, the test case files are present inside a folder named Scenarios. For multiple test cases, corresponding folders should be present.

Outcome:

When test case execution is started, a test suite file per test case containing test details is created automatically in testsuites folder.

Name	Date modified	Type	Size
JIRA-1	03-07-2020 20:26	Microsoft Excel C...	1 KB
JIRA-1	02-07-2020 17:42	XML File	1 KB

JIRA-1.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">
<suite name="TestPlan">
    <test name="JIRA-1">
        <parameter name="xmlFilePath" value="testsuites\Scenarios\JIRA-1\JIRA-1.xml"></parameter>
        <parameter name="csvFilePath" value="testsuites\Scenarios\JIRA-1\JIRA-1.csv"></parameter>
        <classes>
            <class name="com.steepgraph.ta.framework.App" />
        </classes>
    </test>
</suite>
```

The JIRA of type bug would contain updated details related to Test Execution in **Issue Links** section as shown below:

Test Project / JIRA-1

Test Login functionality

Details

Type:	Bug	Status:	TO DO (View Workflow)
Priority:	Medium	Resolution:	Unresolved
Labels:	None		

Description
Click to add description

Attachments

Drop files to attach, or browse.

Issue Links

relates to

- JIRA-8 Test Login functionality = DONE
- JIRA-9 Test Login functionality = DONE
- JIRA-10 Test Login functionality = DONE
- JIRA-11 Test Login functionality = DONE
- JIRA-12 Test Login functionality = DONE

Show 14 more links (10 relates to, 4 tests)

On completion of execution, a JIRA of type Task is created for the Bug issue and its status is reflected accordingly in **Details** section

Test Project / JIRA-8

Test Login functionality

Details

Type:	<input checked="" type="checkbox"/> Task	Status:	DONE (View Workflow)
Priority:	Medium	Resolution:	Done
Labels:	None		

Description
Click to add description

Attachments

Drop files to attach, or browse.

JIRA-1_output.csv
3 days ago 0.1 kB

Issue Links

relates to

- JIRA-1 Test Login functionality = TO DO

The test execution details contain Execution Evidence and Comment. As the test case has passed, only the output csv file has been uploaded to the execution details. If it failed,

a snapshot of the failure would also be uploaded along with a comment in the Comment box.

CSV file for pass case:

username	password	Execution Status	Execution Failure Reason(if any)	Notes(if any)
3dx_user1	T0g3aShU6v47E1iaY3JEaw==	PASS		

On the local side, in the test case folder, the output csv file is created for the passed test case.

Name	Date modified	Type	Size
JIRA-1	03-07-2020 20:26	Microsoft Excel C...	1 KB
JIRA-1	02-07-2020 17:42	XML File	1 KB
JIRA-1_output	03-07-2020 20:44	Microsoft Excel C...	1 KB

In case of failure:

Execution object:

Details		Status:	DONE (View Workflow)
Type:	<input checked="" type="checkbox"/> Task		
Priority:	= Medium	Resolution:	Done
Labels:	None		

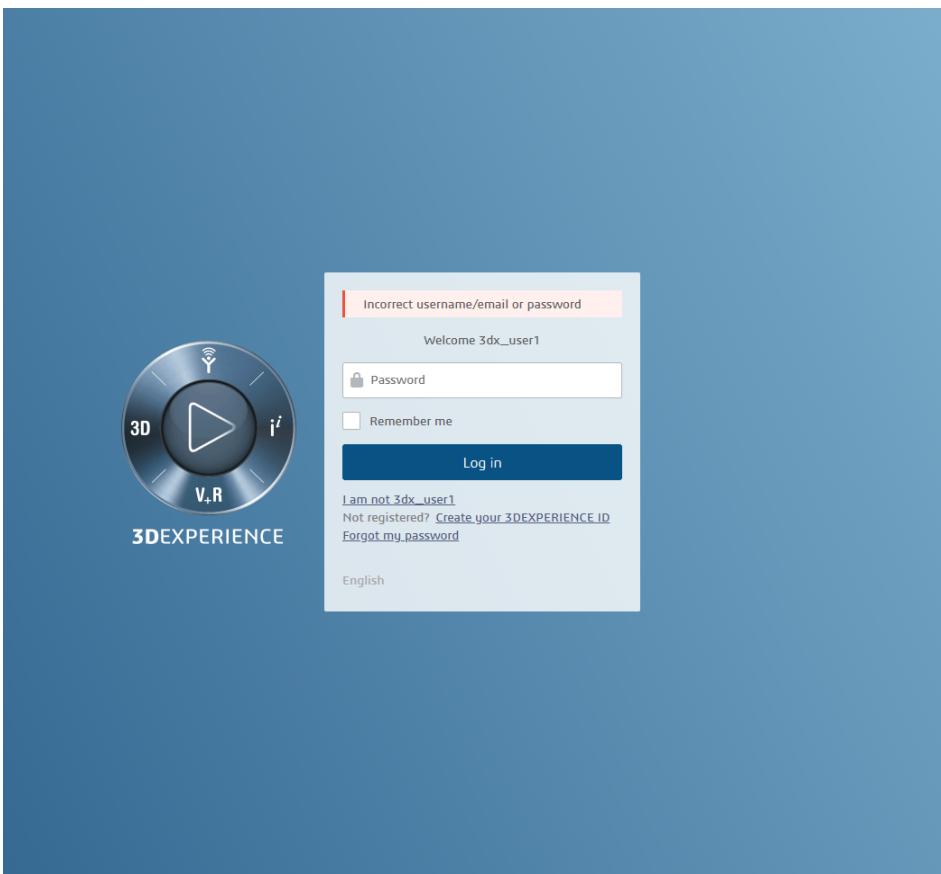
Local folder:

Name	Date modified	Type	Size
Scenarios	03-07-2020 17:17	File folder	
JIRA-1	06-07-2020 11:43	XML File	1 KB
JIRA-19	03-07-2020 20:43	XML File	1 KB
JIRA-20	03-07-2020 20:42	XML File	1 KB

Failure output CSV file:

A	B	C	D	E
username	password	Execution Status	Execution Failure Reason(if any)	Notes(if any)
3dx_user1	0g3aShU6v47E1iaY3JEaw==	FAIL	Keys to send should be a not null CharSequence	

Failure screenshot:



2. Executing New Feature with one or more test cases:

Here New Feature executes functionality of Test Plan(i.e. combinations of different test cases, in order to meet the required criteria) as per values in Jira.properties.

Configuration and execution:

The JIRA issue type for execution is New Feature. Tickets of type Bug in JIRA, corresponding to test cases on the local system to be executed, will be added to this New Feature.

The screenshot shows a JIRA issue page for 'To implement both Login and Part creation' (JIRA-19). The issue is categorized as a 'New Feature' with a priority of 'Medium' and no labels. The status is 'TO DO' with a link to 'View Workflow'. The 'Description' section has a placeholder 'Click to add description'. The 'Attachments' section has a placeholder 'Drop files to attach, or browse.' The 'Issue Links' section shows three links: 'tested by' JIRA-1 (Test Login functionality) and JIRA-20 (To implement Part Creation), both marked as 'TO DO'; and 'tests' JIRA-30 (combinations), also marked as 'TO DO'.

Jira.properties:

```
#Start Execution in Integration
jira.use.integration=true

jira.testsuites.folders = Scenarios

# Test Case/Test Plan/Story Id
Scenarios.jira.object.execute = JIRA-19

jira.test.environment=steepgraphdev
```

In TestProject folder, test cases to be executed under New Feature JIRA-19 are present inside folders named JIRA-1 and JIRA-20, corresponding to Tests JIRA-1 and JIRA-20 respectively.

Name	Date modified	Type
JIRA-1	03-07-2020 20:26	File folder
JIRA-20	03-07-2020 17:17	File folder

Outcome:

When test case execution is started, a test suite file per New Feature containing test details is created automatically in testsuites folder.

Name	Date modified	Type	Size
Scenarios	06-07-2020 14:01	File folder	
JIRA-19	03-07-2020 20:43	XML File	1 KB

JIRA-19.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">
<suite name="TestPlan">
    <test name="JIRA-1">
        <parameter name="xmlFilePath" value="testsuites\Scenarios\JIRA-1\JIRA-1.xml"></parameter>
        <parameter name="csvFilePath" value="testsuites\Scenarios\JIRA-1\JIRA-1.csv"></parameter>
        <classes>
            <class name="com.steepgraph.ta.framework.App" />
        </classes>
    </test>
    <test name="JIRA-20">
        <parameter name="xmlFilePath" value="testsuites\Scenarios\JIRA-20\JIRA-20.xml"></parameter>
        <parameter name="csvFilePath" value="testsuites\Scenarios\JIRA-20\JIRA-20.csv"></parameter>
        <classes>
            <class name="com.steepgraph.ta.framework.App" />
        </classes>
    </test>
</suite>
```

A single JIRA issue of type Task is created for the New Feature and the completion status in **Details** section is reflected accordingly.

The screenshot shows a JIRA issue page for "To implement both Login and Part creation" (JIRA-19). The page includes the following sections:

- Summary:** To implement both Login and Part creation
- Buttons:** Edit, Comment, Assign, More, Start Progress, Done, Admin
- Details:**
 - Type: New Feature
 - Priority: Medium
 - Labels: None
 - Status: TO DO (View Workflow)
 - Resolution: Unresolved
- Description:** Click to add description
- Attachments:** Drop files to attach, or browse.
- Issue Links:**
 - tested by
 - JIRA-1 Test Login functionality
 - JIRA-20 To implement Part Creation
 - tests
 - JIRA-30 combinations

Test Execution(Bug) object:

TestProject / JIRA-24

Test Login functionality

Edit Comment Assign More Reopen Reopen and start progress Admin

Details

Type:	<input checked="" type="checkbox"/> Task	Status:	DONE (View Workflow)
Priority:	<input type="checkbox"/> Medium	Resolution:	Done
Labels:	None		

Description
Click to add description

Attachments

Drop files to attach, or browse.

JIRA-1_output.csv
2 days ago 0.1 kB

Issue Links

relates to

JIRA-1 Test Login functionality = TO DO

+

The screenshot shows a Jira issue page for 'To implement Part Creation'. The top navigation bar includes 'Edit', 'Comment', 'Assign', 'More', 'Reopen', 'Reopen and start progress', and 'Admin'. The issue details are as follows:

- Type:** Task (checked)
- Priority:** Medium
- Status:** DONE (View Workflow)
- Resolution:** Done
- Labels:** None

The 'Description' section contains the placeholder text 'Click to add description'.

The 'Attachments' section shows a single file named 'JIRA-20_output.csv' uploaded 2 days ago, which is 0.3 kB in size. A placeholder text 'Drop files to attach, or browse.' is also present.

The 'Issue Links' section shows a link to another issue: 'JIRA-20 To implement Part Creation' with a status of 'TO DO'.

3. Executing EPIC with test cases and/or test set containing test cases:

Here Epic executes functionality of User Story (It basically is a requirement that a functionality or feature must have written from perspective of end user, so it includes all the requirements and its related test cases which needs to be tested) as per values in Jira.properties.

Configuration and execution:

The JIRA issue type for Epic is Story containing a Improvement with some New Feature and some Bugs.

 TestProject / JIRA-30
combinations

[Edit](#) [Comment](#) [Assign](#) [More](#) [Start Progress](#) [Done](#) [Admin](#)

Details

Type:	 Epic	Status:	 TO DO (View Workflow)
Priority:	 Medium	Resolution:	Unresolved
Labels:	None		
Epic Name:	To test combination of Test		

Description
Click to add description

Attachments

 Drop files to attach, or browse.

Issue Links

tested by

-  JIRA-1 Test Login functionality  TO DO
-  JIRA-18 To Test Part Creation  TO DO
-  JIRA-19 To implement both Login and Part creation  TO DO

Issues in Epic

There are no issues in this epic.

 TestProject / JIRA-18
To Test Part Creation

[Edit](#) [Comment](#) [Assign](#) [More](#) [Start Progress](#) [Done](#) [Admin](#)

Details

Type:	 Improvement	Status:	 TO DO (View Workflow)
Priority:	 Medium	Resolution:	Unresolved
Labels:	None		

Description
Click to add description

Attachments

 Drop files to attach, or browse.

Issue Links

tested by

-  JIRA-1 Test Login functionality  TO DO
-  JIRA-20 To implement Part Creation  TO DO

tests

-  JIRA-30 combinations  TO DO

Activity

All [Comments](#) Work Log History Activity

There are no comments yet on this issue.

```
#Start Execution in Integration
jira.use.integration=true

jira.testsuites.folders = Scenarios

# Test Case/Test Plan/Story Id
Scenarios.jira.object.execute = JIRA-30

jira.test.environment=steepgraphdev
```

Name	Date modified	Type
JIRA-1	03-07-2020 20:26	File folder
JIRA-20	03-07-2020 17:17	File folder

Outcome:

When test case execution is started, a test suite file per Test containing test details is created automatically in testsuites folder. Per test case, an execution object is created and its success/failure is accordingly reflected in JIRA.

Name	Date modified	Type	Size
Scenarios	03-07-2020 17:17	File folder	
JIRA-1	06-07-2020 11:43	XML File	1 KB
JIRA-19	03-07-2020 20:43	XML File	1 KB
JIRA-20	03-07-2020 20:42	XML File	1 KB

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">
<suite name="TestPlan">
    <test name="JIRA-20">
        <parameter name="xmlFilePath" value="testsuites\Scenarios\JIRA-20\JIRA-20.xml"></parameter>
        <parameter name="csvFilePath" value="testsuites\Scenarios\JIRA-20\JIRA-20.csv"></parameter>
        <classes>
            <class name="com.steepgraph.ta.framework.App" />
        </classes>
    </test>
</suite>
```

TestProject / JIRA-35

Test Login functionality

Edit Comment Assign More Reopen Reopen and start progress Admin

Details

Type:	<input checked="" type="checkbox"/> Task	Status:	DONE (View Workflow)
Priority:	<input type="checkbox"/> Medium	Resolution:	Done
Labels:	None		

Description
Click to add description

Attachments

Drop files to attach, or browse.

JIRA-1_output.csv
6 minutes ago 0.2 kB

Issue Links

relates to

JIRA-1 Test Login functionality = TO DO

+

To implement both Login and Part creation

Type: Task Status: DONE (View Workflow)
 Priority: Medium Resolution: Done
 Labels: None

Description
Click to add description

Attachments

Drop files to attach, or browse.

	JIRA-1_output.csv	4 minutes ago	0.2 kB
	JIRA-20_output.csv	4 minutes ago	0.3 kB

Issue Links

relates to

- JIRA-1 Test Login functionality
- JIRA-20 To implement Part Creation

4. Executing a Improvement with Bugs:

Here Improvement executes functionality of TestSet(It is group of different test cases which needs to be executed together) as per values in Jira.properties.

Configuration and execution:

The JIRA issue type for Task is Improvement containing some Bugs

TestProject / JIRA-18

To Test Part Creation

[Edit](#) [Comment](#) [Assign](#) [More](#) [Start Progress](#) [Done](#) [Admin](#)

Details

Type:	Improvement	Status:	TO DO (View Workflow)
Priority:	Medium	Resolution:	Unresolved
Labels:	None		

Description

Click to add description

Attachments

Drop files to attach, or browse.

Issue Links

tested by

- [JIRA-1](#) Test Login functionality **TO DO**
- [JIRA-20](#) To implement Part Creation **TO DO**

tests

- [JIRA-30](#) combinations **TO DO**

Activity

All [Comments](#) Work Log History Activity

There are no comments yet on this issue.

```
#Start Execution in Integration
jira.use.integration=true

jira.testsuites.folders = Scenarios

# Test Case/Test Plan/Story Id
Scenarios.jira.object.execute = JIRA-18

jira.test.environment=steepgraphdev
```

Name	Date modified	Type
JIRA-1	03-07-2020 20:26	File folder
JIRA-20	03-07-2020 17:17	File folder

Outcome:

When test case execution is started, a test suite file per Test containing test details is created automatically in testsuites folder. Per test case, an execution object is created and its success/failure is accordingly reflected in JIRA.

Name	Date modified	Type	Size
Scenarios	06-07-2020 14:01	File folder	
JIRA-1	06-07-2020 14:13	XML File	1 KB
JIRA-20	06-07-2020 14:13	XML File	1 KB

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">
<suite name="TestPlan">
    <test name="JIRA-1">
        <parameter name="xmlFilePath" value="testsuites\Scenarios\JIRA-1\JIRA-1.xml"></parameter>
        <parameter name="csvFilePath" value="testsuites\Scenarios\JIRA-1\JIRA-1.csv"></parameter>
        <classes>
            <class name="com.steepgraph.ta.framework.App" />
        </classes>
    </test>
</suite>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">
<suite name="TestPlan">
    <test name="JIRA-20">
        <parameter name="xmlFilePath" value="testsuites\Scenarios\JIRA-20\JIRA-20.xml"></parameter>
        <parameter name="csvFilePath" value="testsuites\Scenarios\JIRA-20\JIRA-20.csv"></parameter>
        <classes>
            <class name="com.steepgraph.ta.framework.App" />
        </classes>
    </test>
</suite>
```

A single JIRA issue of type Task is created for the New Feature and the completion status in **Details** section is reflected accordingly.

Test Execution/Issue(Bug) object:

The screenshot shows a JIRA issue page for a task titled 'Test Login functionality'. The page includes the following sections:

- Header:** Buttons for Edit, Comment, Assign, More, Reopen, Reopen and start progress, and Admin.
- Details:** Fields for Type (Task), Priority (Medium), Labels (None), Status (DONE), and Resolution (Done).
- Description:** A placeholder text 'Click to add description'.
- Attachments:** A file named 'JIRA-1_output.csv' attached 2 days ago, weighing 0.1 kB.
- Issue Links:** A 'relates to' link to another JIRA issue 'JIRA-1 Test Login functionality'.

TestProject / JIRA-25

To implement Part Creation

[Edit](#) [Comment](#) [Assign](#) [More](#) [Reopen](#) [Reopen and start progress](#) [Admin](#)

Details

Type:	<input checked="" type="checkbox"/> Task	Status:	DONE (View Workflow)
Priority:	<input type="checkbox"/> Medium	Resolution:	Done
Labels:	None		

Description
Click to add description

Attachments

Drop files to attach, or browse.

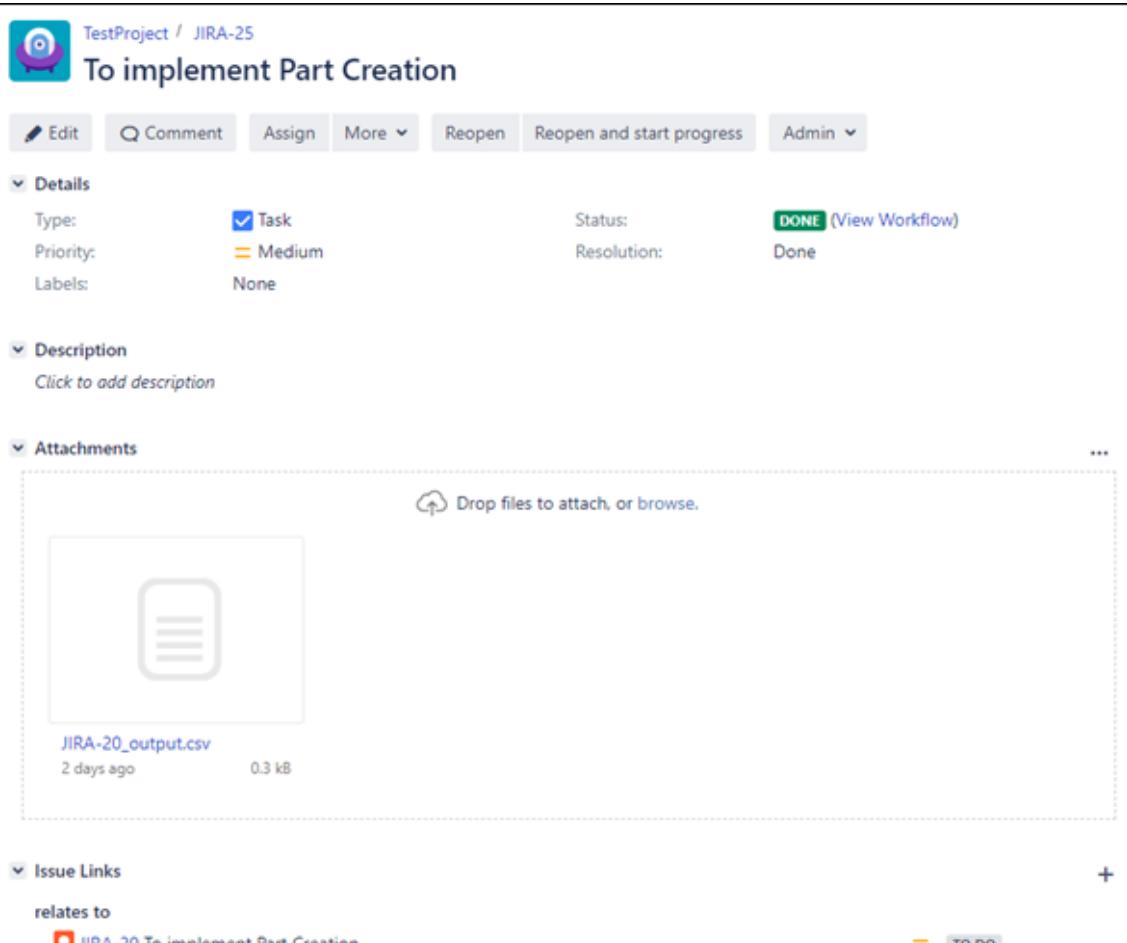
JIRA-20_output.csv
2 days ago 0.3 kB

Issue Links

relates to

JIRA-20 To implement Part Creation TO DO

+



In case, any execution gets failed, the execution object for the same would contain screenshot as well in the attachment, as provided below:

Test Project / JIRA-47

Test Login functionality

Details

Type: Task Priority: Medium Labels: None Status: TO DO (View Workflow) Resolution: Unresolved

Description
Click to add description

Attachments

Drop files to attach, or browse.

	JIRA-1_1.png Yesterday 768 kB		JIRA-1_output.csv Yesterday 0.2 kB
--	-------------------------------------	--	--

Issue Links

relates to

JIRA-1 Test Login functionality TO DO

Created with the Personal Edition of HelpNDoc: [Free HTML Help documentation generator](#)

JIRA Execution Result

General checks before executing Test Plan with JIRA XRAY Integration:

1. The folder specified by `jira.xray.testsuites.folders` key should be present in `testsuites` folder in 3DX-TAS. There can be multiple folders in `testsuites` for testing. Accordingly the property values can be comma separated.
Example:
`jira.xray.testsuites.folders=Folder1,Folder2`
2. In the property `folder.jira.xray.object.execute`, folder must be replaced by the appropriate folder name from the `testsuites` folder.
Example:
`Folder1.jira.xray.object.execute`
3. The value of the key `folder.jira.xray.object.execute` should match the issue ID of the JIRA test/testplan/Story to be executed.
Example:
`Folder1.jira.xray.object.execute = TES-11`, where TES-11 is the JIRA issue ID
4. There can be multiple instances of the key `folder.jira.xray.object.execute` in case of multiple

folders.

Example:

If there are two folders, folder1 and folder 2

Folder1.jira.xray.object.execute = TES-11

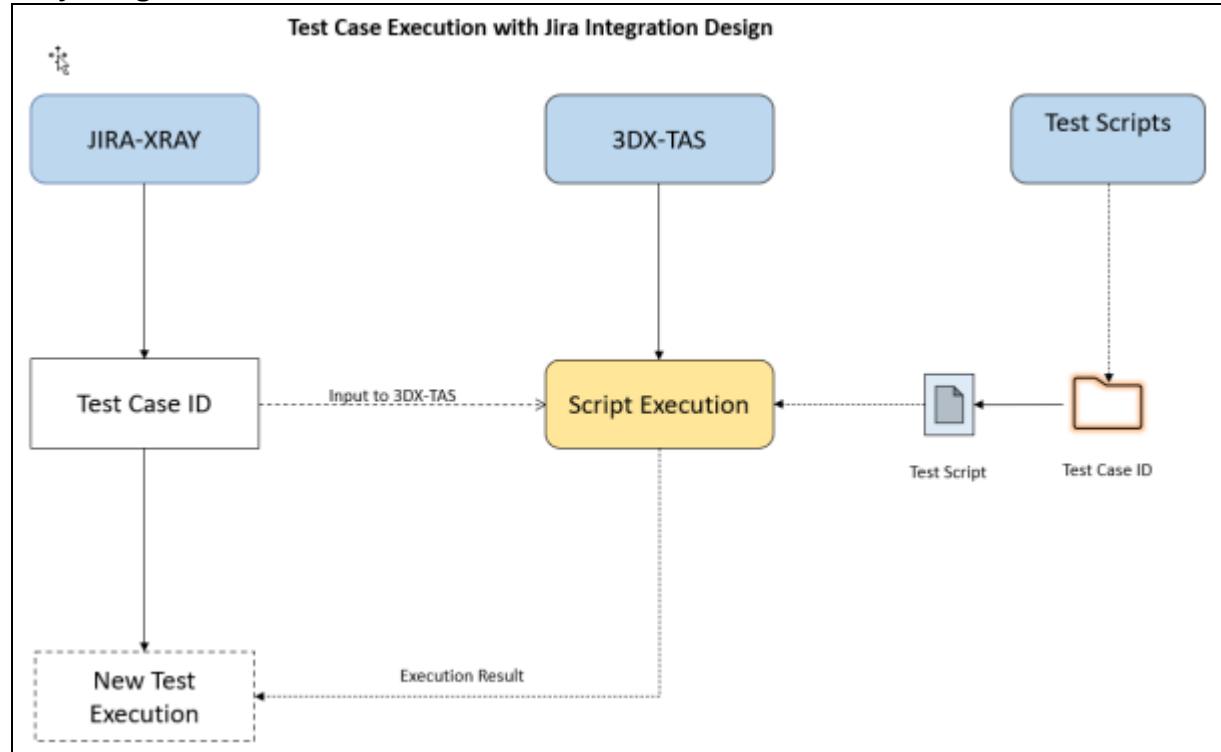
Folder2.jira.xray.object.execute = TES-22

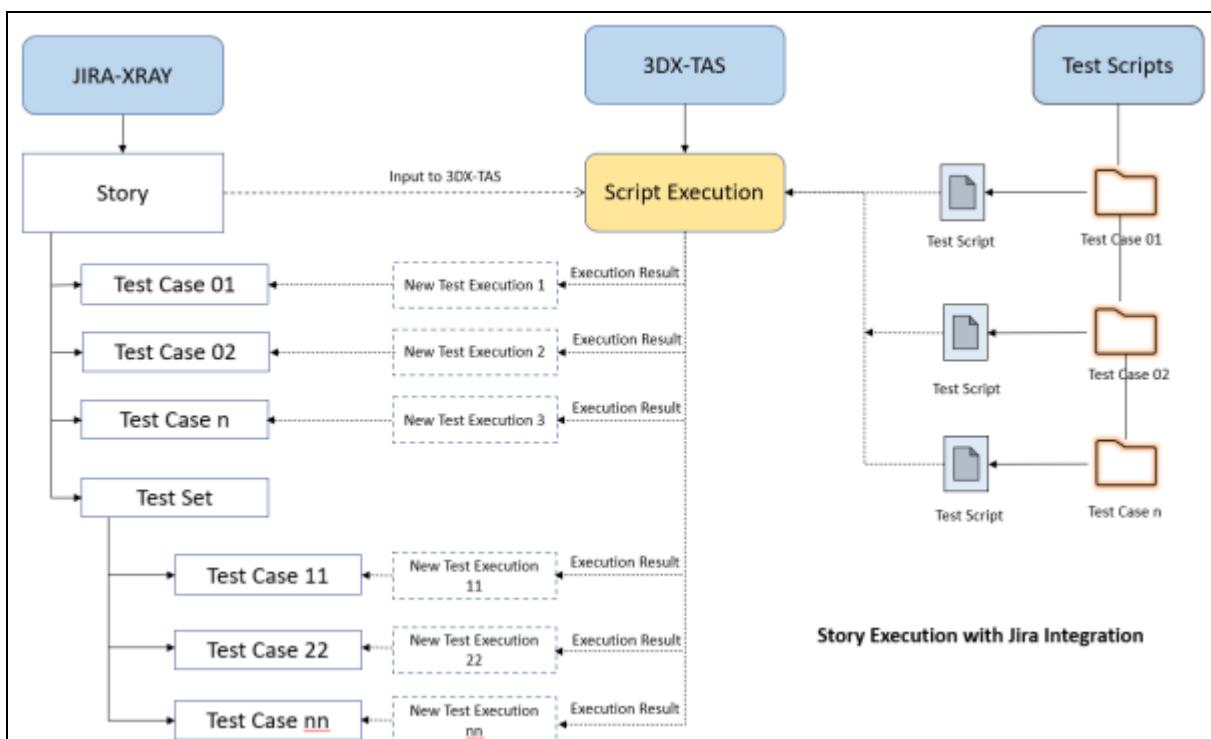
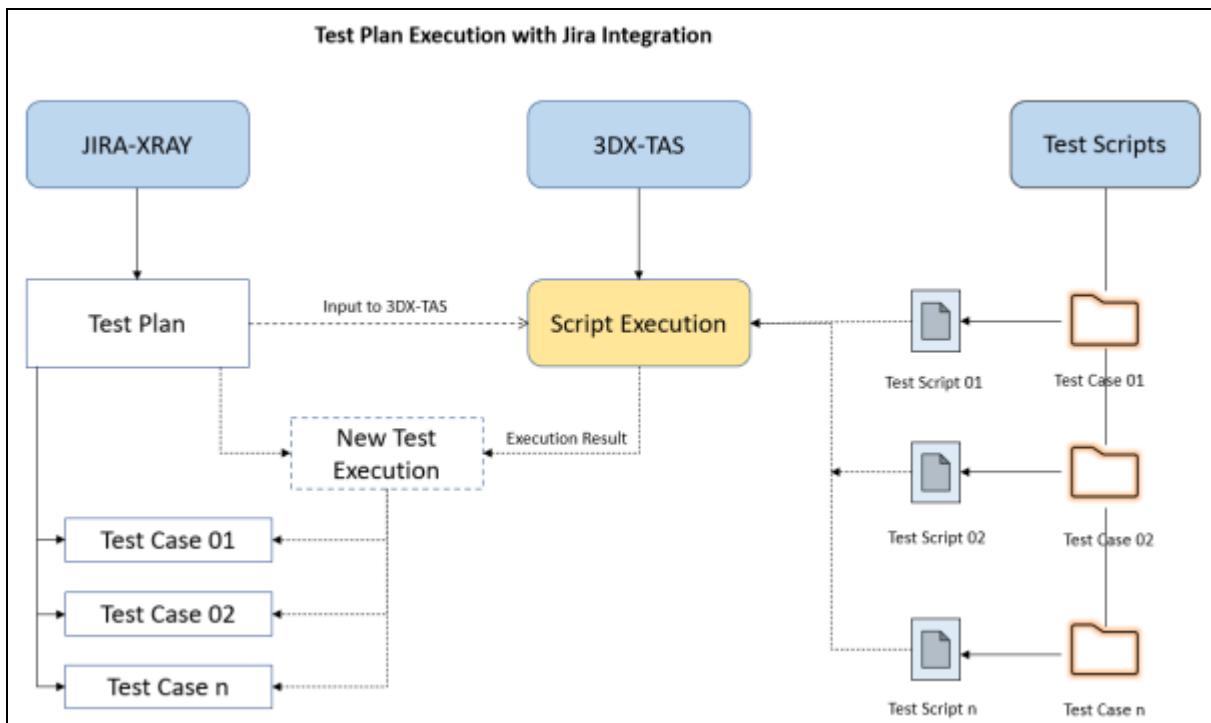
Xray Integration Test Scenarios:

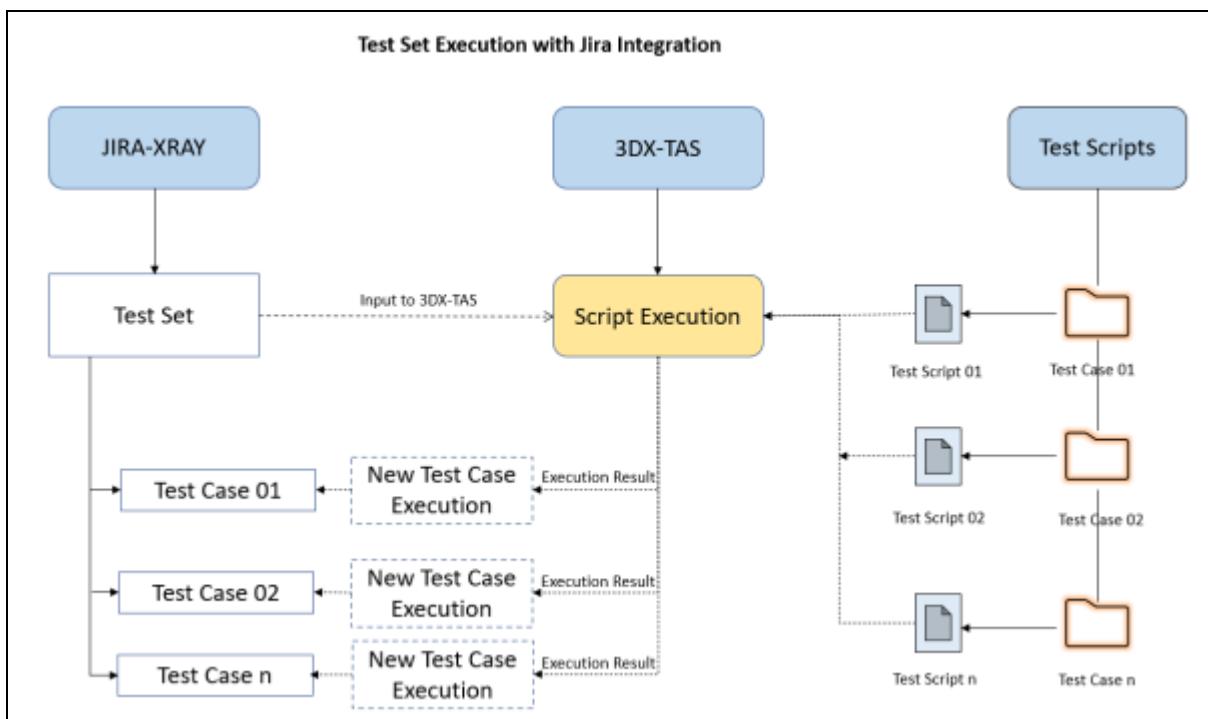
There are multiple scenarios in which the Xray integration can be used.

- Executing one or more test cases.
- Executing test plan with one or more test cases.
- Executing Story with test cases and/or test set containing test cases.
- Executing a test set with test cases.

Xray Integration flow scenarios:







Examples:

1. For explaining the execution scenarios let us consider a project "TAS Testing Project" in JIRA. It has been configured to allow Xray coverage and it's issue types also include Xray issue types.
 2. In TAS local setup, testsuites folder contains a folder named TASTestingProject in which all the relevant test cases will be present.
 3. The key `jira.xray.testsuites.folders` has been set to `jira.xray.testsuites.folders=TASTestingProject` in `XRAY.properties` file.
 4. `folder.jira.xray.object.execute` key will be set to appropriate values based on JIRA issue ID(s) being tested.
 5. On completion of test, the Test Execution object created for the test will be set to Done state specified by the property `jira.xray.testExecution.promotion=41`. The process to set the state is described in the Configuration section for 3DX-TAS.

1. Executing one or more test cases:

Configuration and execution:

The JIRA issue type for execution is Test. If more than one test case is to be executed, corresponding tickets must exist in the JIRA project.

TAS Testing Project / TAS-1

Test 1 - Create Part

[Edit](#) [Comment](#) [Assign](#) [More ▾](#) [To Do](#) [In Progress](#) [Workflow ▾](#) [Admin ▾](#)

Details

Type:	<input checked="" type="radio"/> Test	Status:	TO DO (View Workflow)
Priority:	<input type="radio"/> Medium	Resolution:	Unresolved
Labels:	None		

Description

This is a test to be executed.

XRAY.properties:

```
#JIRA Xray and 3DX-TAS Integration
#-----
#Start Execution in Integration
jira.xray.use.integration=true

jira.xray.testsuites.folders = TASTestingProject

# Test Case/Test Plan/Story Id
TASTestingProject.jira.xray.object.execute = TAS-1

jira.xray.test.environment=steepgraphdev
```

For multiple Tests, comma separated values can be given for the property TASTestingProject.jira.xray.object.execute.

Example:

```
# Test Case/Test Plan/Story Id
TASTestingProject.jira.xray.object.execute = TAS-1, TAS-2
```

In TASTestingProject folder, the test case files are present inside a folder named TAS-1. For multiple test cases, corresponding folders should be present.

Name	Date modified	Type	Size
TAS-1.csv	20-04-2020 15:47	Microsoft Excel Co...	1 KB
TAS-1.xml	20-04-2020 19:58	XML File	1 KB

Outcome:

When test case execution is started, a test suite file per test case containing test details is created automatically in testsuites folder.

Name	Date modified	Type	Size
TASTestingProject	22-04-2020 18:19	File folder	
TAS-1.xml	22-04-2020 18:51	XML File	1 KB

TAS-1.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">
<suite name="TestPlan">
  <test name="TAS-1">
    <parameter name="xmlFilePath" value="testsuites\TASTestingProject\TAS-1\TAS-1.xml"></parameter>
    <parameter name="csvFilePath" value="testsuites\TASTestingProject\TAS-1\TAS-1.csv"></parameter>
    <classes>
      <class name="com.steepgraph.ta.framework.App" />
    </classes>
  </test>
</suite>
```

On completion of execution, a JIRA issue of type Test Execution is created for the Test issue and its pass or fail status is reflected accordingly.

The screenshot displays two JIRA pages related to test management.

Top Page (Test 1 - Create Part):

- Header:** [Project] TAS-2 | TAS-1 Test 1 - Create Part
- Toolbar:** Edit, Comment, Assign, More, To Do, In Progress, Workflow, Admin
- Form Fields:**
 - Type: Test (selected), Priority: Medium, Labels: None
 - Status: Unresolved (with View Workflow link)
 - Description: This is a test to be executed.
 - Test Details: Xray Custom Fields Not Configured (Note: This section cannot be displayed because some required custom fields are not configured. Please make sure this project is using an 'Issue Type Screen Scheme' which contains the Xray Screen Scheme. In order for Xray custom fields to be editable, they must be configured for this issue type's CREATE/EDIT screen.)
 - Pre-Conditions: This test is not associated with Pre-Conditions yet. (Buttons: Create Pre-Condition, Associate Pre-Conditions)
 - Test Sets: This test is not associated with Test Sets yet. (Buttons: Associate Test Sets)
 - Test Plans: This test is not associated with Test Plans yet. (Buttons: Associate Test Plans)
 - Test Runs: (Table)

Project	Version (project dependent)	Status	Start	End
All Projects	Select a project to enable		DD-MM-YYYY HH:MM	DD-MM-YYYY HH:MM

 Show 50 entries Columns ▾
- Bottom Page (Test Execution History):**

Test Execution History Table:

Key	Fix Version/s	Revision	Executed By	Started	Finished	Defects	Status
TAS-2			suchitg@steepgraph.com	2 minutes ago	Just now		PASS

Show 10 entries Columns ▾

Showing 1 to 1 of 1 entries First Previous 1 Next Last

Test Execution object created in JIRA. Its status is set to Done post execution irrespective of success/failure of the test case:

The screenshot shows the TAS Testing Project interface with the following details:

- Project:** TAS Testing Project / TAS-2
- Test Case:** Test 1 - Create Part
- Type:** Test Execution
- Priority:** Lowest
- Status:** DONE (View Workflow)
- Resolution:** Done
- Description:** This is a test to be executed.
- Tests:** Overall Execution Status: 1 PASS
- Test Details:**

Rank	Key	Summary	Test Type	#Req	#Def	Test Sets	Assignee	Status
1	TAS-1	Test 1 - Create Part	Manual	0	0		suchitg@steepgraph.com	PASS

The details of execution can be seen from the Test Execution object as shown below:

The screenshot shows the Test Execution object details page with the following information:

- Key:** TAS-2
- Executed By:** suchitg@steepgraph.com
- Started:** 3 minutes ago
- Finished:** 1 minute ago
- Status:** PASS
- Execution Details:** EXECUTE INLINE, TODO, EXECUTING, FAIL, ABORTED
- Attachments:** Drop files to attach or browse.
- Activity:**

The screenshot shows the Test Execution object details page with the following information:

- Test Case:** Test 1 - Create Part
- Comments:** By pressing **ctrl + v**, you can attach a screenshot from clipboard to Execution Evidence. (Demo)
- Evaluation License:** This is an evaluation license that will expire on 2020-05-07 19:30. If you like Xray, please consider buying it.
- Execution Status:** PASS
- Started On:** 22/Apr/20 6:51 PM
- Finished On:** 22/Apr/20 6:53 PM
- Assignee:** suchitg@steepgraph.com
- Version:** -
- Executed By:** suchitg@steepgraph.com
- Revisions:** -
- Tests environments:** -
- Comment:** Click to add comment.
- Execution Defects (0):** Create Defect · Create Sub-Task · Add Defects
- Execution Evidence (1):** Add Evidence
- File:** TAS-1_output.csv (0.2 kB, 2 minutes ago)

The test execution details contain Execution Evidence and Comment. As the test case has passed, only the output csv file has been uploaded to the execution details. If it failed,

a snapshot of the failure would also be uploaded along with a comment in the Comment box.

CSV file for pass case:

A	B	C	D	E	F	G	H
password	username	Description	AutoName	Execution	Execution	Notes(if any)	
T0g3aShU(3dx_user1	Test Part	A Series	PASS				

On the local side, in the test case folder, the output csv file is created for the passed test case.

Name	Date modified	Type	Size
TAS-1.csv	20-04-2020 15:47	Microsoft Excel Co...	1 KB
TAS-1.xml	20-04-2020 19:58	XML File	1 KB
TAS-1_output.csv	22-04-2020 18:53	Microsoft Excel Co...	1 KB

In case of failure:

Execution object:

Key	Fix Version/s	Revision	Executed By	Started	Finished	Defects	Status
TAS-3			suchitg@steepgraph.com	16 minutes ago	11 minutes ago		FAIL

TAS Testing Project / Test Execution: TAS-3 / Test: TAS-1

[Return to Test Execution](#)

Test 1 - Create Part

1 By pressing **ctrl + V** you can attach a screenshot from clipboard to Execution Evidence.

[Dismiss](#)

1 Evaluation license
This is an evaluation license that will expire on 2020-05-07 19:59. If you like Xray, please consider buying it.

Execution Status: **FAIL** 

Started On: **22/Apr/20 7:30 PM**  Finished On: **22/Apr/20 7:35 PM**

Assigned To: [suchitg@steepgraph.com](#) 
Executed By: [suchitg@steepgraph.com](#) 
Tests environment: 

Comment	Preview Comment
Input XML: testresultsTASTestingProjectTAS-TAS-1.xml, SuiteTestPlan Test Case:TAS-1: Exception Expected condition failed: waiting for com.steepgraph.ta.framework.common.pages.HD107ed9fc (timed for 30 second(s)) with 5000 milliseconds interval	

Execution Defects (0)	Create Defect	Create Sub-Task	Add Defects
 No defects yet...			

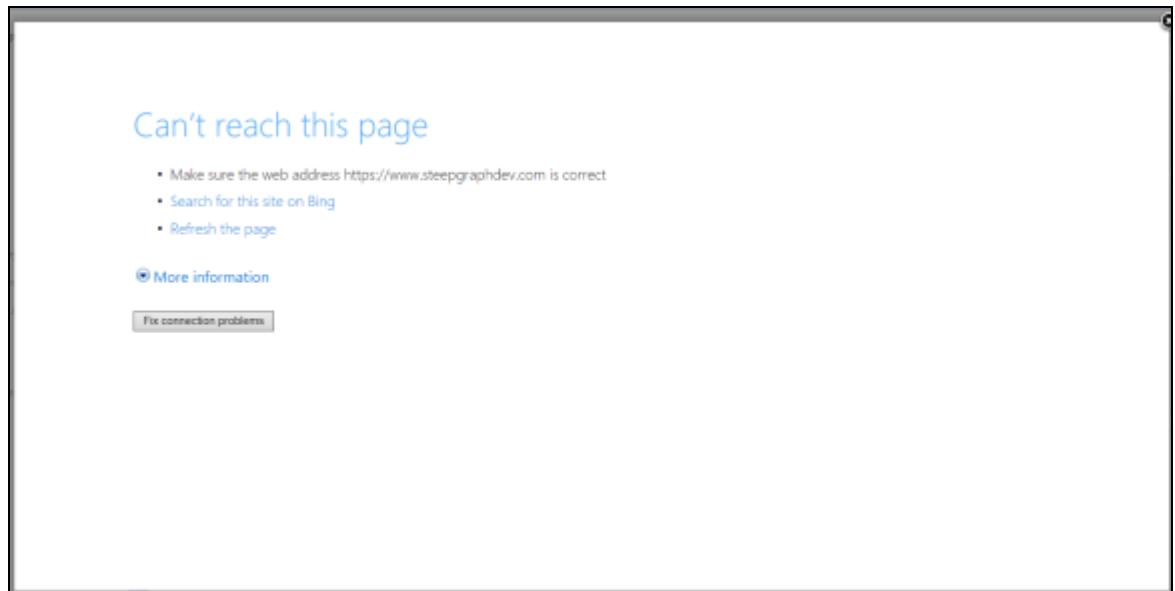
Execution Evidence (2)	Add Evidence
 TAS-1_output.log  TAS-1_1.png	0.3 kB 12 minutes ago 27 kB 12 minutes ago

Local folder:

Name	Date modified	Type	Size
TAS-1.csv	20-04-2020 15:47	Microsoft Excel Co...	1 KB
TAS-1.xml	22-04-2020 19:30	XML File	1 KB
TAS-1_1.png	22-04-2020 19:35	PNG File	28 KB
TAS-1_output.csv	22-04-2020 19:35	Microsoft Excel Co...	1 KB

Failure output CSV file:

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
password	username	Description	AutoName	Execution	Execution	Notes(if any)												
T0g3a5hU13dx_user1	Test Part	A-Series		FAIL		Expected condition failed: waiting for com.steepgraph.ta.framework.common.pages.f@107ed6fc (tried for 30 second(s) with 5000 milliseconds interval)												

Failure screenshot:**2. Executing test plan with one or more test cases:****Configuration and execution:**

The JIRA issue type for execution is Test Plan. Tickets of type Test in JIRA, corresponding to test cases on the local system to be executed, will be added to this Test Plan.

Key	Summary	Requirements	#Test Executions	Issue Assignee	Latest Status	Actions
TAS-5	Test Case 5		0	suchitg@steepgraph.com	TODO	...
TAS-6	Test Case 6		0	suchitg@steepgraph.com	TODO	...

XRAY.properties:

```
#JIRA Xray and 3DX-TAS Integration
-----
#Start Execution in Integration
jira.xray.use.integration=true

jira.xray.testsuites.folders = TASTestingProject

# Test Case/Test Plan/Story Id
TASTestingProject.jira.xray.object.execute = TAS-4

jira.xray.test.environment=steepgraphdev
```

In TASTestingProject folder, test cases to be executed under Test Plan TAS-4 are present inside folders named TAS-5 and TAS-6, corresponding to Tests TAS-5 and TAS-6 respectively.

Name	Date modified	Type	Size
TAS-5	23-04-2020 16:39	File folder	
TAS-6	23-04-2020 16:57	File folder	

Outcome:

When test case execution is started, a test suite file per Test Plan containing test details is created automatically in testsuites folder.

Name	Date modified	Type	Size
TASTestingProject	23-04-2020 16:57	File folder	
TAS-4.xml	23-04-2020 17:31	XML File	1 KB

TAS-4.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">
<suite name="TestPlan">
    <test name="TAS-5">
        <parameter name="xmlFilePath" value="testsuites\TASTestingProject\TAS-5\TAS-5.xml"></parameter>
        <parameter name="csvFilePath" value="testsuites\TASTestingProject\TAS-5\TAS-5.csv"></parameter>
        <classes>
            <class name="com.steepgraph.ta.framework.App" />
        </classes>
    </test>
    <test name="TAS-6">
        <parameter name="xmlFilePath" value="testsuites\TASTestingProject\TAS-6\TAS-6.xml"></parameter>
        <parameter name="csvFilePath" value="testsuites\TASTestingProject\TAS-6\TAS-6.csv"></parameter>
        <classes>
            <class name="com.steepgraph.ta.framework.App" />
        </classes>
    </test>
</suite>
```

A single JIRA issue of type Test Execution is created for the Test Plan and the pass or fail status is reflected accordingly.

TAS Testing Project / TAS-4

Test Plan 1

[Edit](#) [Comment](#) [Assign](#) [More](#) [To Do](#) [In Progress](#) [Workflow](#) [Admin](#)

Details

Type:	<input checked="" type="checkbox"/> Test Plan	Status:	TO DO (View Workflow)
Priority:	<input type="checkbox"/> Medium	Resolution:	Unresolved
Labels:	None		

Description

Tests

[Test Plan Board](#) [+ Create Test Execution](#) [+ Add](#)

Overall Execution Status

2 PASS

Total Tests: 2

[Filter\(s\)](#)

Key	Summary	Requirements	#Test Executions	Issue Assignee	Latest Status	Actions
TAS-5	Test Case 5		1	suchitg@steepgraph.com	PASS	...
TAS-6	Test Case 6		1	suchitg@steepgraph.com	PASS	...

Showing 1 to 2 of 2 entries [First](#) [Previous](#) [Next](#) [Last](#)

TAS Testing Project / TAS-5

Test Case 5

[Edit](#) [Comment](#) [Assign](#) [More](#) [To Do](#) [In Progress](#) [Workflow](#) [Admin](#)

Details

Type:	<input checked="" type="checkbox"/> Test	Status:	TO DO (View Workflow)
Priority:	<input type="checkbox"/> Medium	Resolution:	Unresolved
Labels:	None		

Description

Test Details

Pre-Conditions

Test Sets

Test Plans

Test Runs

[Execute In](#)

FILTERS

Project	Version (project dependent)	Status	Start	End
All Projects	Select a project to enable		DD-MM-YYYY HH:MM	DD-MM-YYYY HH:MM

Show 10 entries Columns ▾

Key	Fix Version/s	Revision	Executed By	Started	Finished	Defects	Status
TAS-7			suchitg@steepgraph.com	11 minutes ago	8 minutes ago		PASS

Showing 1 to 1 of 1 entries [First](#) [Previous](#) [Next](#) [Last](#)

Test Execution object:

The screenshot shows the TAS Testing Project interface. At the top, there's a navigation bar with 'TAS Testing Project / TAS-7' and a 'Test Plan 1' title. Below the title are buttons for 'Edit', 'Comment', 'Assign', 'More', 'To Do', 'In Progress', 'Workflow', and 'Admin'. A dropdown menu for 'Details' is open, showing 'Type: Test Execution', 'Priority: Lowest', 'Labels: None', 'Status: Done', and a link '(View Workflow)'. Under 'Description', it says 'Overall Execution Status' with a green bar indicating 2 PASS. The 'Tests' section shows a table with two entries:

Rank	Key	Summary	Test Type	#Req	#Def	Test Sets	Assignee	Status
2	TAS-6	Test Case 6	Manual	0	0		suchitg@steepgraph.com	PASS
1	TAS-5	Test Case 5	Manual	0	0		suchitg@steepgraph.com	PASS

At the bottom, it says 'Showing 1 to 2 of 2 entries' and has links for 'First', 'Previous', 'Next', and 'Last'.

Execution details for TAS-5:

This screenshot shows the execution details for Test Case 5. It includes sections for 'Execution Evidence' (with a note about attaching screenshots), 'Evaluation license' (an evaluation license that expires on 2020-03-07 19:30), and 'Execution Status' (PASS). The execution timeline is from 23/Apr/20 5:31 PM to 23/Apr/20 5:34 PM. On the right, it shows assignment to 'suchitg@steepgraph.com', version '1', and revision '1'. The execution summary table contains three rows: 'Comment' (Click to add comment), 'Execution Defects (0)' (No defects yet...), and 'Execution Evidence (1)' (A file named 'TAS-5.csv' uploaded 0.2 kB 7 minutes ago).

3. Executing Story with test cases and/or test set containing test cases: Configuration and execution:

The JIRA issue type for execution is Story containing a Test Set with some test cases and some independent test cases.

 TAS Testing Project / TAS-8
Story 1

[Edit](#) [Comment](#) [Assign](#) [More](#) [To Do](#) [In Progress](#) [Workflow](#) [Admin](#)

Details

Type:	Story	Status:	TO DO (View Workflow)
Priority:	Medium	Resolution:	Unresolved
Labels:	None		

Description
This is a Story containing a Test Set and independent Test Cases for execution.

Attachments

Drop files to attach, or browse.

Issue Links

tested by

 TAS-12	Test 12	 TO DO
 TAS-13	Test 13	 TO DO
 TAS-9	Test Set 1	 TO DO

 TAS Testing Project / TAS-9
Test Set 1

[Edit](#) [Comment](#) [Assign](#) [More](#) [To Do](#) [In Progress](#) [Workflow](#) [Admin](#)

Details

Type:	Test Set	Status:	TO DO (View Workflow)
Priority:	Medium	Resolution:	Unresolved
Labels:	None		

Description
This is a Test Set containing Test cases for execution.

Tests

Evaluation license
This is an evaluation license that will expire on 2020-05-07 19:30. If you like Xray, please consider buying it.

[Create Test](#) [Add Tests](#)

Key	Summary	Actions
1 TAS-10	Test 10	...
2 TAS-11	Test 11	...

Show 10 entries Columns

Showing 1 to 2 of 2 entries First Previous **1** Next Last

```
#JIRA Xray and 3DX-TAS Integration
#-----
#Start Execution in Integration
jira.xray.use.integration=true

jira.xray.testsuites.folders = TASTestingProject

# Test Case/Test Plan/Story Id
TASTestingProject.jira.xray.object.execute = TAS-8

jira.xray.test.environment=steepgraphdev
```

Name	Date modified	Type	Size
TAS-10	23-04-2020 18:45	File folder	
TAS-11	23-04-2020 18:45	File folder	
TAS-12	23-04-2020 18:45	File folder	
TAS-13	23-04-2020 18:45	File folder	

Outcome:

When test case execution is started, a test suite file per Test containing test details is created automatically in testsuites folder. Per test case, an execution object is created and its success/failure is accordingly reflected in JIRA.

Name	Date modified	Type	Size
TASTestingProject	23-04-2020 18:44	File folder	
TAS-10.xml	23-04-2020 19:06	XML File	1 KB
TAS-11.xml	23-04-2020 19:07	XML File	1 KB
TAS-12.xml	23-04-2020 19:03	XML File	1 KB
TAS-13.xml	23-04-2020 19:04	XML File	1 KB

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">
<suite name="TestPlan">
  <test name="TAS-12">
    <parameter name="xmlFilePath" value="testsuites\TASTestingProject\TAS-12\TAS-12.xml"></parameter>
    <parameter name="csvFilePath" value="testsuites\TASTestingProject\TAS-12\TAS-12.csv"></parameter>
    <classes>
      <class name="com.steepgraph.ta.framework.App" />
    </classes>
  </test>
</suite>
```

The screenshot shows the JIRA interface for the 'TAS Testing Project / TAS-12' issue. The top navigation bar includes 'Edit', 'Comment', 'Assign', 'More', 'To Do', 'In Progress', 'Workflow', and 'Admin'. Below the header, there's a 'Details' section with fields for Type (Test), Priority (Medium), Labels (None), Status (TO DO), and Resolution (Unresolved). A 'Description' section contains the text: 'This is a Test Case for execution.' Below this are sections for 'Test Details', 'Pre-Conditions', 'Test Sets', 'Test Plans', and 'Test Runs'. A 'FILTERS' section allows selecting a project, setting start and end dates, and choosing columns. The 'Test Runs' table shows one entry for 'TAS-14' with status 'PASS'. At the bottom, a message indicates that pressing **ctrl + v** allows attaching screenshots from clipboard to Execution Evidence.

Project	Version (project dependent)	Status	Start	End
All Projects	Select a project to enable		DD-MM-YYYY HH:MM	DD-MM-YYYY HH:MM

Execution Status: **PASS**

Started On: 23/Apr/20 7:53 PM Finished On: 23/Apr/20 7:54 PM

Assigned: suchitg@steepgraph.com Version: -
Executed By: suchitg@steepgraph.com Revision: -
Tests environments: -

Comment: Click to add comment

Execution Defects (0): Create Defect | Create Sub-Task | Add Defects
No defects yet...

Execution Evidence (1): Add Evidence
TAG-12_output.csv (0.2 kB) 14 minutes ago

4. Executing a test set with test cases:

Configuration and execution:

The JIRA issue type for execution is Test Set containing some test cases

 TAS Testing Project / TAS-18

Test Set 2

[Edit](#) [Comment](#) [Assign](#) [More](#) [To Do](#) [In Progress](#) [Workflow](#) [Admin](#)

[Details](#)

Type:	<input checked="" type="checkbox"/> Test Set	Status:	TO DO (View Workflow)
Priority:	<input type="checkbox"/> Medium	Resolution:	Unresolved
Labels:	None		

[Description](#)

This is a Test Set containing Test Cases for execution.

[Tests](#)

Evaluation license

This is an evaluation license that will expire on 2020-05-07 19:30. If you like Xray, please consider [buying it](#).

[Create Test](#) [Add Tests](#)

Key	Summary	Columns
1 TAS-19	Test 19	...
2 TAS-20	Test 20	...

Showing 1 to 2 of 2 entries

First Previous **1** Next Last

```
#JIRA Xray and 3DX-TAS Integration
#-----
#Start Execution in Integration
jira.xray.use.integration=true

jira.xray.testsuites.folders = TASTestingProject

# Test Case/Test Plan/Story Id
TASTestingProject.jira.xray.object.execute = TAS-18
jira.xray.test.environment=steepgraphdev
```

Name	Date modified	Type	Size
TAS-19	23-04-2020 19:25	File folder	
TAS-20	23-04-2020 19:25	File folder	

Outcome:

When test case execution is started, a test suite file per Test containing test details is created automatically in testsuites folder. Per test case, an execution object is created and its success/failure is accordingly reflected in JIRA.

Name	Date modified	Type	Size
TASTestingProject	23-04-2020 19:24	File folder	
TAS-19.xml	23-04-2020 19:36	XML File	1 KB
TAS-20.xml	23-04-2020 19:37	XML File	1 KB

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">
<suite name="TestPlan">
  <test name="TAS-19">
    <parameter name="xmlFilePath" value="testsuites\TASTestingProject\TAS-19\TAS-19.xml"></parameter>
    <parameter name="csvFilePath" value="testsuites\TASTestingProject\TAS-19\tas-19.csv"></parameter>
    <classes>
      <class name="com.steepgraph.ta.framework.App" />
    </classes>
  </test>
</suite>
```

 TAS Testing Project / TAS-19

Test 19

[Edit](#) [Comment](#) [Assign](#) [More](#) [To Do](#) [In Progress](#) [Workflow](#) [Admin](#)

Details

Type:	<input checked="" type="radio"/> Test	Status:	TO DO (View Workflow)
Priority:	<input checked="" type="radio"/> Medium	Resolution:	Unresolved
Labels:	None		

Description
This is a Test case for execution.

Test Details

Pre-Conditions

Test Sets

Test Plans

Test Runs

[Execute In](#)

FILTERS

Project	Version (project dependent)	Status	Start	End
All Projects	Select a project to enable		DD-MM-YYYY HH:MM	DD-MM-YYYY HH:MM

Show 10 entries Columns ▾

Key	Fix Version/s	Revision	Executed By	Started	Finished	Defects	Status
TAS-21			suchitg@steepgraph.com	4 minutes ago	3 minutes ago	PASS	View

Showing 1 to 1 of 1 entries First Previous [1](#) Next Last

The screenshot shows the XRAY interface for a test execution named 'Test 19'. At the top, there's a note about attaching screenshots via keyboard shortcut (ctrl + v). Below it, an 'Evaluation License' section indicates an expiration date of 2020-05-07 19:33. The execution status is 'PASS' with a green bar. The execution details show it started at 23/Apr/20 7:36 PM and finished at 23/Apr/20 7:37 PM. There are no comments or defects. One piece of evidence, 'TAS-19_output.csv', is listed.

Selective Execution of Test Cases based on Status of last execution:

The test cases under a TestSet or TestPlan can be executed selectively based on status of last execution of TestCase.

Under XRAY.properties:

```

35
36 jira.xray.execute.selective=true
37 jira.xray.execute.selective.status=fail
38 #possible values:todo,pass,fail,aborted

```

If `jira.xray.execute.selective` is set to true, then the test cases under a TestSet or TestPlan which have their last execution status same as that provided in `jira.xray.execute.selective.status` are only executed from the list of test cases.

Created with the Personal Edition of HelpNDoc: [Modernize your help files with HelpNDoc's WinHelp HLP to CHM conversion tool](#)

TAG</> Library

All TAG Available in Alphabetical wise A TO Z:

There are total 3Types of Tag Library :
 1.Web Library (Includes basic web and 3dx Tag)
 2.Native Library (Includes Native 3dx & BasicTags Only)
 3.Arás Library (Includes Arás tags only)

Created with the Personal Edition of HelpNDoc: [Maximize Your Documentation Efficiency with a Help Authoring Tool](#)

Web

This is custom xml tag library is defined in SteepGraph test automation tool. Each tag is a specific action to perform on web browser. Let's go in the details of each tag.

Created with the Personal Edition of HelpNDoc: [Save time and frustration with HelpNDoc's WinHelp HLP to CHM conversion feature](#)

Action

Action:

Perform a given action on web element. Web Element should be located using find-element tag first.

NOTE: id attribute of FindElement tag will be used as refid for this tag.

Attribute name defines the action to be performed on given web element.

- **Syntax:**

<Action name="" refid="" />

- **Attributes:**

- **refid:(Mandatory)**

- id of FindElement tag will used as refid for this tag.

- **name:(Mandatory)** It is a mandatory attribute. Here are the possible values of attribute name:

- **click:**

Clicks on given web element. Functionally action tag and [ClickElement](#) are similar, but technically they are different.

For some cases ClickElement tag is not able click on web element, in such cases use action tag.

- **doubleclick:**

Double click on the given web element.

- **rightclick:**

It's mouse right button on given web element.

- **input:**

This is like [InputText](#) tag. User need to pass extra id attribute and define input data in csv file.

<Action name="input" refid="" value="LEFT_CONTROL"/>

- **clickandhold:**

Clicks (without releasing) at the current mouse location .

- **hover:**

This attribute is used when user want to hover on certain element. with cursor pointer.

- **release:**

Releases the depressed left mouse button at the current mouse location.

- **mousedown:**

This attribute is used for MouseDown action with findElement tag and it's alternative for DranAndDrop tag. (Drag means mouseDown Oration)

It's click the **mouse Right button for Down action.**

- **mouseup:**

This attribute is used for MouseUp action with findElement tag and it's alternative for DranAndDrop tag. (Drop means mouseUp Oration)

It's click the **mouse Right button for UP action.**

- **keydown:**

Performs a modifier key press after focusing on an element. Does not release the modifier key - subsequent interactions may assume it's kept pressed.

User need to pass name of the key to press in csv file in the same way we define for other tags.

- **Example:**

Suppose you want to press shift key down then use action tag as mentioned below:

<Action name="keydown" refid="DoneBtn" value="LEFT_CONTROL"/>

Please check below the key names defined by selenium.:

NULL	SPACE	NUMPAD0	F1
CANCEL	PAGE_UP	NUMPAD1	F2
HELP	PAGE_DOWN	NUMPAD2	F3
BACK_SPACE	END	NUMPAD3	F4
TAB	HOME	NUMPAD4	F5
CLEAR	LEFT	NUMPAD5	F6
RETURN	ARROW_LEFT	NUMPAD6	F7
ENTER	UP	NUMPAD7	F8
SHIFT	ARROW_UP	NUMPAD8	F9
LEFT_SHIFT	RIGHT	NUMPAD9	F10
CONTROL	ARROW_RIGHT	MULTIPLY	F11
LEFT_CONTROL	DOWN	ADD	F12
ALT	ARROW_DOWN	SEPARATOR	
LEFT_ALT	INSERT	SUBTRACT	
PAUSE	DELETE	DECIMAL	
ESCAPE	SEMICOLON	DIVIDE	

- **keyUp:**

Performs a modifier key release after focusing on an element.

Examples:

Suppose you want to release SHIFT key which is pressed earlier in the given test case then use action tag action mentioned below:

```
<Action name="keyup" refid="DoneBtn" id="" "KeyName" "SHIFT" "/>
```

- **value:(Optional)**

- this attribute used when user chose input ,keyup , keydown as name attribute , then this attribute is mandatory.Possible Values are modifier_key for keyup and keydown.

For keyup , keydown possible values are given below.

NULL	SPACE	NUMPAD0	F1
CANCEL	PAGE_UP	NUMPAD1	F2
HELP	PAGE_DOWN	NUMPAD2	F3
BACK_SPACE	END	NUMPAD3	F4
TAB	HOME	NUMPAD4	F5
CLEAR	LEFT	NUMPAD5	F6
RETURN	ARROW_LEFT	NUMPAD6	F7
ENTER	UP	NUMPAD7	F8
SHIFT	ARROW_UP	NUMPAD8	F9
LEFT_SHIFT	RIGHT	NUMPAD9	F10
CONTROL	ARROW_RIGHT	MULTIPLY	F11
LEFT_CONTROL	DOWN	ADD	F12
ALT	ARROW_DOWN	SEPARATOR	
LEFT_ALT	INSERT	SUBTRACT	
PAUSE	DELETE	DECIMAL	
ESCAPE	SEMICOLON	DIVIDE	

- **Examples:**

- ```
<FindElement locatorType="cssselector" locatorExpression=".btn-primary" id="DoneBtn"/>
<Action name="click" refid="DoneBtn"/>
```

```

<SwitchToFrame refid="sourcelframe1"/>
<!-- Fd06 Object -->
<FindElement locatorType="xpath" locatorExpression="//div[@class='wux-tweakers
wux-tweakers-urlobject']//a[@class='wux-tweakers-string-label' and text()='DOC-
89686518-0000211']" id="m1"/>
<Action name="mousedown" refid="m1"/>
<Wait time="1000" />
<FindElement locatorType="xpath" locatorExpression="//div[@class='wp-tab-panel
selected']//iframe[contains(@src,'ENOLCMI_AP')]" id="tolframe2"/>
<SwitchToFrame refid="tolframe2"/>
<Wait time="10000" />
<FindElement locatorType="xpath"
locatorExpression="//div[@class='searchView']//div[@class='sentence-details align-
center']/button" id="m2"/>
<Action name="mouseup" refid="m2"/>

```

- Note : Alternatives for Drag and Drop tag, If it is dragAndDrop tag is not working, For your reference please check below code.

- Iframes in Drag and Drop Objects

```

<FindElement locatorType="xpath" locatorExpression="//div[@class='wp-tab-panel
selected']//div[@class='moduleContent']//iframe[contains(@src,'ENXWDOC_AP')]"
id="sourcelframe1"/>

```

- Without Iframes in Drag and Drop Objects

```

<FindElement locatorType="xpath" locatorExpression="//div[@data-id='ENORERE_AP']"
id="m1"/>
<Wait time="1000" />
<Action name="mousedown" refid="m1"/>
<Wait time="2000" />
<FindElement locatorType="xpath"
locatorExpression="(//div[contains(@class,'moduleWrapper ')])[1]" id="m2"/>
<Action name="mouseup" refid="m2"/>

```

---

Created with the Personal Edition of HelpNDoc: [Add an Extra Layer of Security to Your PDFs with Encryption](#)

## Assert

### Assert:

User can use assert for validation. The Assertion results are based on the comparison of Actual & Expected Results. Use either id or input attribute for this tag

- **Syntax:**

```
<Assert locatorType="id" locatorExpression="" id="" criteria="text" condition="="/>
```

- **Attributes:**

- **locator Type:(mandatory when refid is not used)**

There are 4 types of locator: id, CSSselector, name and xpath which are already discussed in ClickElement tag.

- **locatorExpression:(mandatory when refid is not used)**

This attribute contains expression specific to locator type.

- **id: (optional if input used)**

id attribute is mapped to csv file column header. Input value should be

taken from csv file.

- **criteria:(mandatory)**

possible values are 'text', 'numberOfCharacters' and 'attribute', On which criteria user wants to perform assertion needs to give in this.

- **attribute:(optional)**

This attribute contains expression specific to locator type. If we put criteria = attribute , then we have to put the attribute value accordingly, like we have to put that attribute whose value we want to assert.

use dropdown as attribute value if the asserted value is in selectDropdown tag of that html & also contain dissabled field in the dom.

- **condition:(mandatory)**

There are 8 types of condition: =, !=, <, >, <=, >=, contains, not contains

- **input:(optional when id used )**

Attribute is used to provide a hard-coded value for input instead of using id attribute from csv file.

- **errormessage:(optional)**

User can give hard coded message in this attribute.

- **highlight:(optional)**

This attribute is used to choose whether to highlight the element when working on it. Takes values true or false.

- **refid:(optional)**

id attribute of FindElement tag will used as refid for this tag.

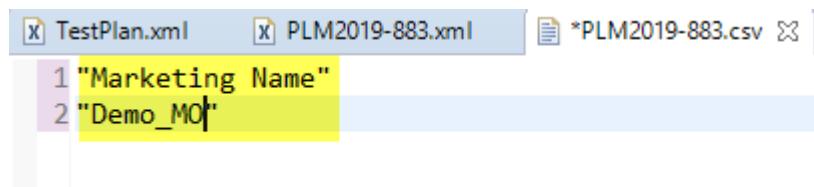
- **style:(optional)**

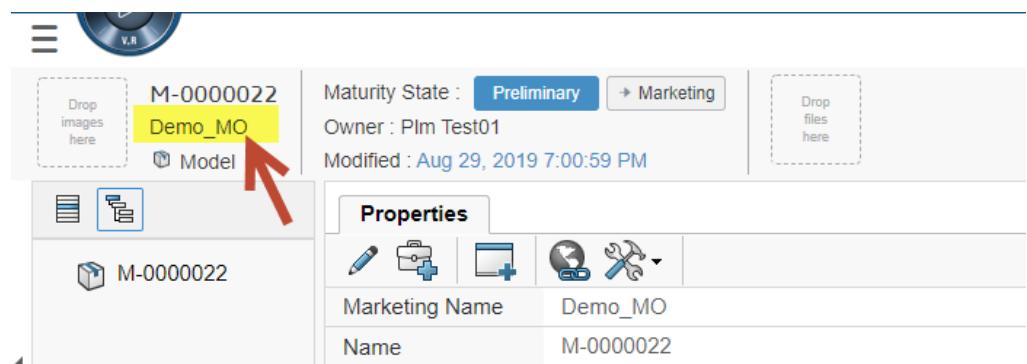
This attribute is used to provide styling to the highlight used.

- **Examples:**

In the image below the user wants to validate the created object. Assert will be compared element text with csv input file(Value of id="Marketing Name" column)

```
<Assert
 locatorExpression="//span[@class='extendedHeader'
 id='Marketing Name' criteria='text' condition='='/>
 locatorType="xpath"
 marketing-name']"
```





- **Examples: condition !=**

In the image below the user wants to validate the created object. Assert will be compared element text with csv input file(Value of id="Marketing Name" column)

```
<Assert locatorType="xpath"
locatorExpression="//span[@class='extendedHeader marketing-name']"
id="Marketing Name" criteria="text" condition="!="/>
```

- **Note**

for above condition our test case will fail ,but actually it is passed for the given condition.

- **Examples: condition = contains**

```
<Assert locatorExpression="//span[@class='extendedHeader type-name']"
id="CheckPart" condition="contains" criteria="text" locatorType="xpath"
highlight="true"/>
```

A	B	C	D	E	F
1 password	username	CheckPart	CheckPart1	State	check
2 T0g3aShU	3DEXP01	Part (A)	Part-0000249-01	In Work	150
3					

**Examples: condition =not contains**

```
<Assert locatorExpression="//span[@class='extendedHeader type-name']"
id="CheckPart1" condition="not contains" criteria="text" locatorType="xpath"
highlight="true"/>
```

	A	B	C	D	E	F	G
1	password	username	CheckPart	CheckPart1	State	check	
2	T0g3aShU	3DEXP01	Part (A)	Part-0000249-01	In Work	150	
3							

The screenshot shows the 3DEXPERIENCE ENOVIA Engineering BOM Management application. At the top, there's a navigation bar with icons for 3D, VR, and other tools. Below it is a search bar with the placeholder "Search" and a magnifying glass icon. A red callout box highlights the search results for "Part-0000249-01". The main workspace displays a part detail view for "Part-0000267-01 A". The properties panel shows the part is in "In Work" status, owned by "3D EXP01", and modified on "2022 Oct 12 18:14:01". The "Properties" tab is selected, showing basic information like Type (Part) and Name (Part-0000267-01). Other tabs include Compliance, Images, Revisions, Customer Extension, and History.

#### Examples: criteria: attribute

```
<Assert locatorExpression="//td[contains(text(),'State')]" id="check"
condition="=" criteria="attribute" attribute="width" locatorType="xpath"
highlight="true"/>
```

Properties   Compliance   Images   Revisions   Customer Extensio...   History

**Basics**

Type	Part		
Name	Part-0000267-01		
Title	Part-0000267-01	Structure Content	Stand Alone
Description			
Revision	A	Phase	Development
State	In Work	Collaborative Policy	EC Part
Classification Path	Part Family		
Derived From			
Creation Date	Oct 12, 2022	Modification Date	Oct 12, 2022

About 3DEXPERIENCE platform

```

<tr id="calc_V_Name"></tr>
<tr id="calc_Description"></tr>
<tr id="calc_Revision"></tr>
<tr id="calc_State">
 <td class="label" width="150" ">>State</td>
 <td width="260.5" class="field">In&nbs;Work&nbs;</td>
 <td class="label" width="150" ">Collaborative Policy</td>
 <td class="field" width="260.5">EC Part&nbs;</td> == $0
</tr>
<tr id="calc_Classification Path"></tr>

```

### Examples: criteria: &gt; greater than

#### NOTES:

in this width value =150

csv file id = check= 100

so the criteria work for this tag is= 100<150 that means width value is greater than the given value in csv file.

```

<Assert locatorExpression="//td[contains(text(),'State')]" id="check"
condition=">" criteria="attribute" attribute="width" locatorType="xpath"
highlight="true"/>
<Assert locatorExpression="//td[contains(text(),'State')]" id="check"
condition=">=" criteria="attribute" attribute="width" locatorType="xpath"
highlight="true"/>

```

	A	B	C	D	E	F
1	password	username	CheckPart	CheckPart	State	check
2	T0g3aShU	3DEXP01	Part (A)	Part-0000	In Work	100
3						

```

▶ <tr id="calc_V_Name">...</tr>
▶ <tr id="calc_Description">...</tr>
▶ <tr id="calc_Revision">...</tr>
▼ <tr id="calc_State">
 <!-- //XSSOK -->
 <td class="label" width="150" >State</td>
 <td width="260.5" class="field">In Work </td>
 <!-- //XSSOK -->
 - <td class="label" width="150" >Collaborative Policy</td>
 <td class="field" width="260.5">EC Part </td> == $0
 </tr>
▶ <tr id="calc_Classification_Path">...</tr>

```

Created with the Personal Edition of HelpNDoc: [Effortlessly Create Professional Documentation with HelpNDoc's Clean UI](#)

## AssertAlert

### AssertAlert :

User can use AssertAlert either to validate the alert messages or to check the presence of alerts.

- **Syntax:**

*<AssertAlert id="" action="" />*

- **Attributes:**

- **id:(optional if action is 'present' or 'notpresent')**

Id attribute is used to compare the alert messages. The value could be taken from the CSV by mentioning its column identifier.

It is only mandatory, in case, **action** attribute value is either **accept** or **dismiss**.

- **action:(mandatory)**

In case user only wants to check the presence of alerts, he/she could use **action** as either present or not present.

The possible values for this attribute are

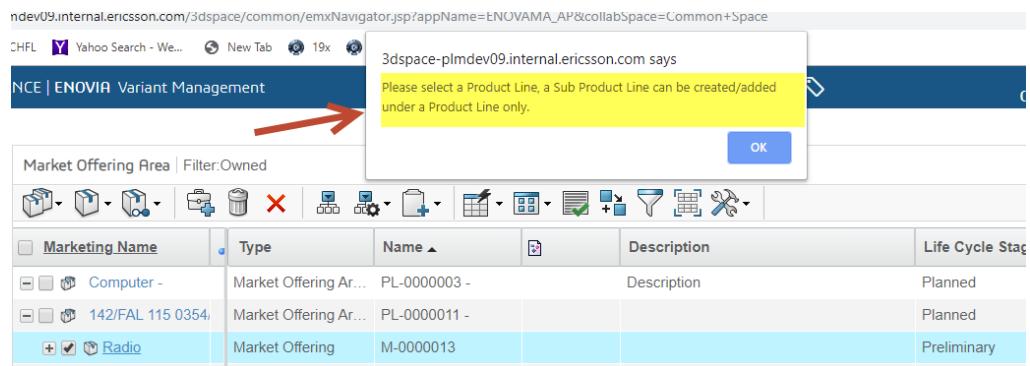
- accept,
- dismiss,
- present,
- notpresent.

- **errormessage: (Optional)** User can give hard coded message in this attribute.

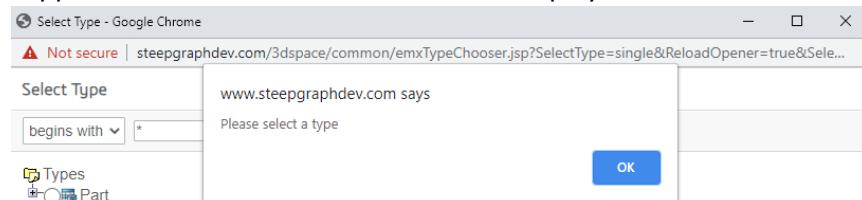
- **Examples:**

In the image below, the user wants to add market offering area under market offering but, error message will be displayed. The tag in this case will be as follows:

*<AssertAlert id="Error message for MOA" action="accept"/>*



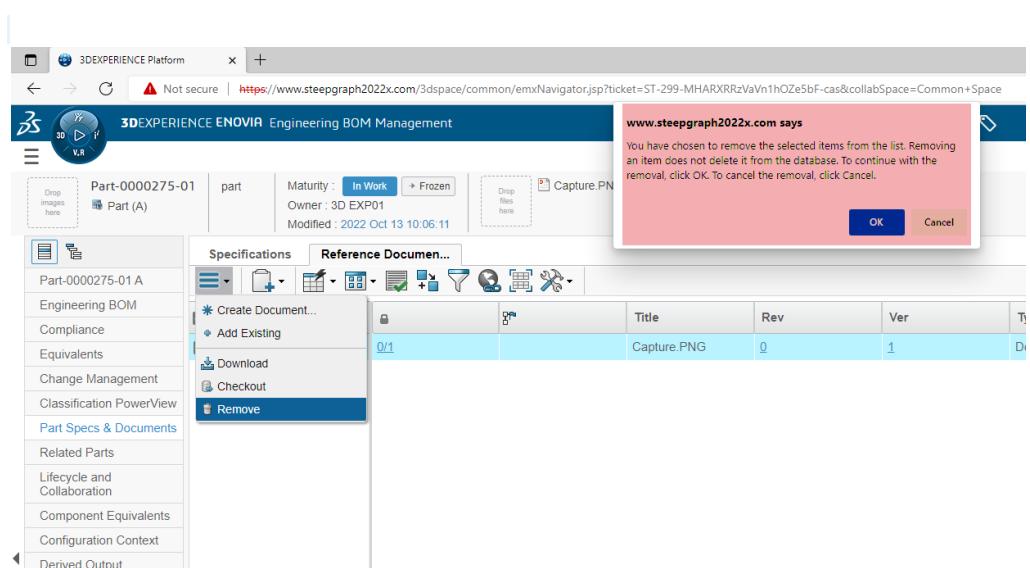
Suppose user wants to validate the alert displayed in the screenshot below:



The syntax for the same is provided below:

`<AssertAlert action="present"/>`

- **Examples: action:dismiss**
- `<AssertAlert id="Error message for MOA" action="dismiss"/>`



**Note:action for this( dismiss=not present), (Accept=present)**

**Examples:**

*<AssertAlert id="Error message for MOA" action="accept" errormessage=" You have chosen to remove the selected specification(s) from the list. Specifications may be deleted based on the behavior of the specification type. To continue with the removal, click OK. To cancel the removal, click Cancel." />*

**www.steepgraph2022x.com says**

You have chosen to remove the selected specification(s) from the list. Specifications may be deleted based on the behavior of the specification type. To continue with the removal, click OK. To cancel the removal, click Cancel.

OK

Cancel

---

Created with the Personal Edition of HelpNDoc: [5 Reasons Why a Help Authoring Tool is Better than Microsoft Word for Documentation](#)

---

## AssertDeletion

**Assertdeletion:**

Assertdeletion tag is used for verifying whether object is removed or deleted from UI.

▪ **Attributes:**

- **locator type :(mandatory)**

There are 4 types of locator: id, cssselector,name and xpath which defines locator type

- **locator Expression:(mandatory)**

This attribute contains Expression specific to locator type .

- **errormessage: (optional)**

User can give hard-coded message in this attribute.

▪ **Syntax:**

*<AssertDeletion locatorType=" " locatorExpression=" "/>*

▪ **Examples:**

*<AssertDeletion locatorType="id" locatorExpression="ECMMYDesk"/>*

*<AssertDeletion locatorType="xpath"*

*locatorExpression="//div[@id='mx\_divBody'] //table//tbody //tr//td[@title='DOC\_50927981\_0000031']"/>*

---

Created with the Personal Edition of HelpNDoc: [Bring your WinHelp HLP help files into the present with HelpNDoc's easy CHM conversion](#)

---

## AssertFileExists

**AssertFileExists:**

User can use AssertFileExists to check file exists on given location.

▪ **Syntax:**

*<AssertFileExists id="">*

▪ **Attributes:**

- **id: (optional if input used)**

Id attribute is csv column identifier. in csv file location is passed.

- **input: (optional if id used)**

attribute is used to provide a hard-coded value of filepath for input instead of using id attribute from csv file.

- **wait: (optional)**

This attribute contains wait time in milliseconds to find element. Default value will be taken from '3dx-tas.execution.step.timeout'. Preferred wait will be based on the time it takes to download the whole file..

▪ **Examples:**

In the image below, the user wants to check downloaded file exist on given location or not. The tag in this case will be as follows:

`<AssertFileExists id="file"/>`

```
1 "file"
2 "C:/NC_Programs/TAS_NC102_TOOL2/This is test file.docx"
```

---

Created with the Personal Edition of HelpNDoc: [Easily create HTML Help documents](#)

## AssertIndentedTableRow

### AssertIndentedTableRow:

AssertIndentedTableRow tag is used for verifying text and attributes of intended table.

▪ **Syntax:**

`<AssertIndentedTableRow position="" refid="" id="" criteria="" condition="" />`

▪ **Attributes:**

- **id (Mandatory):** Id attribute is csv column identifier which is already discussed in InputText tag.
- **refid (Mandatory):** id attribute of [SelectIndentedTableRow](#) tag is the refid of this tag. So, using this refid script will identify the row to be edited.
- **position (Mandatory):** It identifies the column number of the row selected using [SelectIndentedTableRow](#) tag to be edited.

**Examples:**

In the image below, the user wants to edit the Usage column of selected row. Its position element can be found by inspecting the DOM and is found to be 18. So the value of position will be 18 in this case.

The screenshot shows a table with three rows. The first row is expanded to show three sub-rows. The third sub-row has its 'Usage' value ('Standard') highlighted with a red box. Below the table is an Inspector tool displaying the HTML code for the selected cell. The highlighted part of the code is the 'position="18"' attribute.

- criteria (Mandatory):**

On which criteria user wants to perform assertion needs to give in this. Here are possible criteria for AssertIndentedTableRow

#### 1.Text:

It identifies text of any attribute.

#### Examples:

If user wants to verify value of usage column for any row then he needs to give it into CSV for validating.

This screenshot is identical to the one above, showing the same table and Inspector tool. The 'Usage' column for row A-0000151-01 is again highlighted with a red box, and the Inspector tool shows the corresponding HTML code with the 'position="18"' attribute highlighted.

#### 2.attribute:(optional)

It identifies values of inner HTML attribute.

#### Examples:

If user wants to identify value of "Title" attribute of web element then "Attribute" criteria is used.

The screenshot shows a table with two rows. The second row is expanded to show two sub-rows. The second sub-row has its 'State' value ('In Work') highlighted with a yellow box. Below the table is an Inspector tool displaying the HTML code for the selected cell. The highlighted part of the code is the 'title="In Work"' attribute.

- **condition (Mandatory):** There are six conditions user can apply for assertion tag which are as follows:

There are 8 types of condition: =, !=, <, >, <=, >=

**Examples:**

```
<AssertIndentedTableRow position="8" refid="Row" id="title" criteria="text"
condition="=" insideFreezePane="false" highlight="False" />
```

- **errormessage: (optional)**

User can give hardcoded message in this attribute.

**Examples:**

```
<AssertIndentedTableRow position="8" refid="Row" id="title" criteria="text"
condition="=" insideFreezePane="false" highlight="False"
errormessage="chek title from csv" />
```

- **highlight:(optional)**

This attribute is used to choose whether to highlight the element when working on it. Takes values true or false.Default value is false.

**Examples:**

```
<AssertIndentedTableRow position="8" refid="Row" id="title" criteria="text"
condition="=" insideFreezePane="false" highlight="False" />
<AssertIndentedTableRow position="8" refid="Row" id="title" criteria="text"
condition="=" insideFreezePane="false" highlight="true" />
```

- **insideFreezePane:(optional)**

Possible values are 'true' or 'false'.If criteria is true it will check inside the freeze panel.Default value id false.

- **style:(optional)**

This attribute is used to provide styling to the highlight used.Default value is taken from 3DXtAS Property key '`3dx-tas.highlight.webelement.style`'.Only if highlight is true.

**Examples:**

User wants to check state of any part then tag should be as follows:

```
<AssertIndentedTableRow position="8" refid="Test" id="TestText"
criteria="text" condition="="/>
```

Or

User wants to check "Title" HTML attribute then tag should be as follows:

```
<AssertIndentedTableRow position="8" refid="Test" id="TestText"
attribute="title" criteria="attribute" condition="="/>
```

Created with the Personal Edition of HelpNDoc: [Create HTML Help, DOC, PDF and print manuals from 1 single source](#)

## AssertListValues

### AssertListValues:

User can use AssertListValues for validate the values of list box. use either `id` or `input attribute` for this tag

▪ **Syntax:**

```
<AssertListValues locatorType="id" locatorExpression="" id="" criteria="contains"/>
```

▪ **Attributes:**

- **locatorType :(mandatory)**

There are 4 type of locators : ID, cssselectors , name and Xpath which defines locator type.

- **locatorExpression:(mandatory)**

This attribute contains Expression specific to locator type

- **id:(optional when input is used)**

Id attribute is csv column identifier, Input value should be taken from csv file.

- **criteria:(mandatory)**

There are 4 types of criteria: equal, not equal, contains, and not contains

1. **equal** à To verify list values are equal to given input. Values are separated by pipe (|)
2. **not equal** à To verify list values are not equal to given input. Values are separated by pipe (|)
3. **contains** à To verify list values contains given input. Values are separated by pipe (|)
4. **not contains** à To verify list values not contains given input. Values are separated by pipe (|)

- **errormessage: (optional)**

User can give hard coded message in this attribute.

- **highlight: (optional)**

This attribute is used to choose whether to highlight the element when working on it. Takes values true or false.Default value is false.

- **input: (optional when id is used)**

This attribute is used to provide a hard-coded value for input instead of using id attribute from csv file.

- **style: (optional)**

This attribute is used to provide styling to the highlight used.Default value is taken from 3DXtAS Property key '3dx-tas.highlight.webelement.style'. Only if highlight is true.

▪ **Examples:**

In the image below, the user wants to validate the values of drop down list. The tag in this case will be as follows:

```
1 "E-Result","ExecutionResul","comments","file","succ
2 "Completed|Completed With Issues|Failed","Failed",
```

```
<AssertListValues locatorType="id" locatorExpression="ExecutionResult0" id="E-Result" criteria="equal"/>
```

▪ **Examples:**

In the image below, the user wants to validate the values of drop down list. The tag in this case will be as follows:

<AssertListValues locatorType="id" locatorExpression="AutoNameSeriesId" criteria="not equal " highlight="true" input="M Series" />

The screenshot shows the 'Create Part' dialog with the 'Type' field set to 'Part'. The 'AutoName Series' dropdown is highlighted in yellow and contains the value 'A Series'. Below the dialog is a table with columns 'password', 'username', and 'AutoNameSeries'. The 'password' column contains 'T0g3aShU6v47E1iaY3JEaw=='. The 'username' column contains '3DEXP01'. The 'AutoNameSeries' column contains 'Not Selected|A Series|B Series|C Series|D Series|General Series'. The 'A Series' option is highlighted in red.

A	B	C	D	E	F	G	H	I
password	username	AutoNameSeries						
T0g3aShU6v47E1iaY3JEaw==	3DEXP01	Not Selected A Series B Series C Series D Series General Series						

<AssertListValues locatorType="id" locatorExpression="AutoNameSeriesId" criteria="contains" id="AutoNameSeries"/>

The screenshot shows the 'Create Part' dialog with the 'Type' field set to 'Part'. The 'AutoName Series' dropdown is highlighted in yellow and contains the value 'A Series'. Below the dialog is a table with columns 'password', 'username', and 'AutoNameSeries'. The 'password' column contains 'T0g3aShU6v47E1iaY3JEaw=='. The 'username' column contains '3DEXP01'. The 'AutoNameSeries' column contains 'Not Selected|A Series|B Series|C Series|D Series|General Series'. The 'A Series' option is highlighted in red.

A	B	C	D	E	F	G	H	I
password	username	AutoNameSeries						
T0g3aShU6v47E1iaY3JEaw==	3DEXP01	Not Selected A Series B Series C Series D Series General Series						

- not contains** To verify list values not contains given input. Values are separated by pipe (|)
- input** This attribute is used to provide a hard-coded value for input instead of using id attribute from csv file.

<AssertListValues locatorType="id" locatorExpression="AutoNameSeriesId" criteria="not contains" input="M Series" />

password	username	AutoNameSeries T0g3aShUi3DEXP01	Not Selected   A Series   B Series   C Series   D Series   General Series	AutoNameSeries1 M Series
----------	----------	------------------------------------	---------------------------------------------------------------------------	-----------------------------

Create Part

- highlight:**  
This attribute is used to choose whether to highlight the element when working on it. Takes values true or false.

```
<AssertListValues locatorType="id" locatorExpression="AutoNameSeriesId"
criteria="not contains" highlight="true" input="M Series" />
```

Created with the Personal Edition of HelpNDoc: [Effortlessly Spot and Fix Problems in Your Documentation with HelpNDoc's Project Analyzer](#)

## AssertOutlookMailHeader

### AssertOutlookMailHeader:

With the help of this tag, AssertOutlookHeader tag is used to validate email\_id from outlook , to email\_id, Cc email\_Id and subject.

- Syntax:**

```
<AssertOutlookMailHeader validateMailHeader="" subject="" validateMsgString="" />
```

#### NOTES:

- This tag use to Search Outlook header name from Search box

- Attributes:**

- validateMailHeader(optional):**

*This attribute is Option. this attribute from is used validate from id, to email id, cc, to.*

- **subject(optional):**  
This attribute is Option. this attribute subject is used to validate subject from outlook mail..
- **validateMsgString(optional):**  
This attribute is Option. this attribute validateMsgString is used to validate msg body with pipe separated String value

■ **Examples:**

Suppose you want to Assert outlook mail header

```
<AssertOutlookMailHeader validateMailHeader="Bhumit JAVIA/Vishal BHOSALE"
subject="Request for Material Declaration of Parts to [ERICSSON]" validateMsgString ="KDU 137 925/31
R1B"/>
```

---

Created with the Personal Edition of HelpNDoc: [Free HTML Help documentation generator](#)

## AssertPageLoadTime

### AssertPageLoadTime:

With the help of this tag, user could assert the timing of Page Load(i.e. how much time it took to load the page).

■ **Syntax:**

```
<AssertPageLoadTime id="CreatePart" condition="=" input="300" />
```

■ **Attributes:**

● **Id :(Mandatory)**

With help of id attribute, user is able to identify the assert operation. There can be multiple assert page load time in single test case.

it will not take value from CSV file, id is used to make identity the assert operation.

Please check example : <AssertPageLoadTime id="login" input="443" condition="=" />

```
<AssertPageLoadTime id="createRoute"
input="443" condition="=" />
```

● **condition(Mandatory)**

There are 6 types of condition: [= , !=, <, <= , > , >=]

'<' means '<'

'<=' means '<='

'>' means '>'

'>=' means '>='

● **input:(Mandatory)**

This attribute specifies the value in milliseconds against which condition will be evaluated.

---

Created with the Personal Edition of HelpNDoc: [Easily create Help documents](#)

## AssertDate

### AssertDate :

With the help of this tag, User can validate two different date.

- **Syntax:**

```
<AssertDate toCompare="days" dateDifference="0" criteria="=" locatorType="" locatorExpression="" dateFormat="MMM dd, yyyy" date2LocatorType="" date2LocatorExpression="" date2Format="MMM dd, yyyy"/>
<AssertDate toCompare="days" dateDifference="0" criteria="=" id="" dateFormat="MMM dd, yyyy" date2LocatorType="" date2LocatorExpression="" />
<AssertDate toCompare="days" dateDifference="0" criteria="=" input="" dateFormat="MMM dd, yyyy" date2LocatorType="" date2LocatorExpression="" />
<AssertDate toCompare="days" dateDifference="0" criteria="=" locatorType="" locatorExpression="" dateFormat="MMM dd, yyyy" date2Input="" />
<AssertDate toCompare="days" dateDifference="0" criteria="=" locatorType="" locatorExpression="" dateFormat="MMM dd, yyyy" date2Id="" />
<AssertDate toCompare="days" dateDifference="0" criteria="=" dateFormat="MMM dd, yyyy" date2Format="MMM dd, yyyy" date2Input="" input="" />
<AssertDate toCompare="days" dateDifference="0" criteria="=" dateFormat="MMM dd, yyyy" date2Format="MMM dd, yyyy" date2Input="" input="" />
<AssertDate toCompare="days" dateDifference="0" criteria="=" date2Input="" input="" />
```

- **Attributes:**

- **toCompare(Mandatory)**

toCompare attribute is used to mention what to compare. Attribute possible values are mins, hours, days, months, years.

- **dateDifference(Mandatory):**

in dateDifference attribute user have to mention the different between two date 3DX-TAS 2.0 114 / 333 like 1, 2, 3.... as per the to compare attribute

- **criteria(Mandatory):**

in this User have to mention the logical operation to compare the two different date. Attribute possible values are =, <=, >=, !=, <, >.

- **locatorType(Optional if id or input is used) :**

There are 4 types of locator: id, CSSselector,name and xpath which are already discussed in ClickElement tag.This is for Date1

- **locatorExpression(Optional if id or input is used) :**

This attribute contains expression specific to locatorType.This is for Date1

- **date2LocatorType(Optional if date2Id or date2Input is used):**

This attribute contains expression specific to locatorType.This is for Date2

- **date2LocatorExpression(Optional if date2Id or date2Input is used):**

There are 4 types of locator: id, CSSselector,name and xpath which are already discussed in ClickElement tag.This is for Date2

- **input(Optional when id is used):**

attribute is used to provide a hard-coded value for input instead of using id attribute from csv file. This is for date 1.

- **id**(Optional when input is used):

`id(CSV)` attribute is mapped to csv file column header. Input value should be taken from csv file. This is for date 1.

- `date2Input`(Optional when `date2Id` is used):

attribute is used to provide a hard-coded value for input instead of using id attribute from csv file. This is for date 2.

- `date2Id`(Optional when `date2Input` is used):

`id(CSV)` attribute is mapped to csv file column header. Input value should be taken from csv file. This is for date 2.

- `dateFormat(Optional):`

User have to provide the date format that is visible on UI. If user is not providing any dateFormat it will take the default dateFormat that is this dd/MM/yyyy. This is for date 1.

- **date2Format(Optional):**

User have to provide the date format that is visible on UI. If user is not providing any dateFormat it will take the default dateFormat that is this dd/MM/yyyy. This is for date 2.

- *Examples:*

*Suppose you want to write some skipped testcases*

```
<TestCase xmlns="https://www.steepgraph.com" xmlns:xsi="http://www.w3.org/2001/XMLSchema-Instance" xsi:schemaLocation="https://www.steepgraph.com .../resources/xsd/TestAutomationFramework.xsd">
<LoginCase>
 <LoginData>
 <username>${csy}</username>
 <password>${csv}[password]</password>
 </LoginData>
 <WaitTime>3000</WaitTime>
 <ScriptContent>
 <GlobalSearch id="aa">
 <OpenSearchResults id="aa" wait="3000">
 <Criteria type="Text" value="aa" />
 <AssertResult toCompare="days" dateDiffDifference="0" criteria="a" locatorType="xpath" locatorExpression="//[@id='extendedHeaderModified']/a" date2Format="yyyy MM dd HH:mm:ss" dateFormat="yyyy MM dd HH:mm:ss" date2LocatorExpression="//[@id='extendedHeaderModified']/a" />
 <AssertResult toCompare="days" dateDiffDifference="2" criteria="a" locatorType="xpath" locatorExpression="//[@id='extendedHeaderModified']/a" date2Format="yyyy MM dd HH:mm:ss" dateFormat="yyyy MM dd HH:mm:ss" id="Date" date2LocatorType="xpath" date2LocatorExpression="//[@id='extendedHeaderModified']/a" />
 </OpenSearchResults>
 </GlobalSearch>
 <ScriptContent>
 <Logout>
 </Logout>
</TestCase>
```

Created with the Personal Edition of HelpNDoc: [Easily create EBooks](#)

# AssertXML

## **AssertXML :**

With the help of this tag, Include another test case XML in current XML

#### ■ *Syntax:*

```
<AssertXML id="" refid="" criteria="none" match="full" xpathExpression="" regexExpr="" deleteFile="true" />
```

## **NOTES:**

1. This tag use to only when CSV is download in downloaded folder due to running time then it will work.
  2. In this tag user should give proper file path you want to validate into the CSV file with

Extension.

*ributes:*

***id(Optional when input is used):***  
*Id/input attribute is mandatory [match = "Full"] when user have to compare two*

*input*(Optional when id is used):

*Hard coded filepath in xml file which they want to compare.*

```
<AssertXML refid="downloadedFile" criteria="exist" match="full" input="testsuites\\Unzip_Folder\\CreateDashboard.xml" deleteFile="false"/>
```

- **refid(Mandatory):**  
*Value is same as id key for XML extracted file, id attribute of ReadXML tag is used as refid for AssertXML tag.*
- **criteria(Optional):**  
*This attribute is used when match is partial, possible values are exist / notexist. Default value is exist.*
- **match(Mandatory):**  
*Either match is full or partial depending on validation of XML..*
- **xpathExpression(Optional):**  
*xpathExpression is mandatory when [match = "partial"] Provide xpath for which value must be retrieve.*
- **regexExpr(Optional):**  
*Provide regexExpr only when xml file contains regular expression. Possible values are true and false. Default value is false.*
- **deletefile(Optional):**  
*Possible values are true and false, Default value is false. True and false value for delete and undelete zip and xml file respectively. This attribute is only applicable when match is full.*

▪ **Examples:**

*Suppose you want to validate header list from CSV file.*

```
<AssertXML refid="downloadedFile" criteria="exist" match="partial"
xpathExpression="//ProductID[@manufacturerItemNumber='${csv{manufacturerItemNumber}}']"/>
```

*Suppose you want to validate header list from CSV file. It will first unzip zip file and then validate that xpath location that you have given in*

*Assert file location.*

```
<ReadXML unziplocation="C:\\Users\\SGSPC022\\Downloads\\CreateDashboard\\"
unzip="true" id="downloadedFile" fromBrowserDownloadLocation="true" regexExpr="true"
downloadFileExt="zip" deleteFile="false"/>
```

```
<AssertXML refid="downloadedFile" criteria="exist"
match="fullxpathExpression="//OpenCompassApp[@appname='3DPlay']"/>
```

*We have to open that particular file in chrome browser and than we have to inspect that file and find xpath that we have given AssertXML tags as shown in example and also in below*

*image.*

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<TestCase xmlns="https://www.steepgraph.com" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="https://www.steepgraph.com
.../.../resources/xsd/TestAutomationFramework.xsd">
<Login username="$csv{username}" password="$csv{password}" />
<SwitchToDefaultContent/>
<setStartwindow/>
<OpenCompassApp quadrant="east" appname="3DDashboard"/>
<SwitchToWindow/>
<Wait time="1000"/>
<SwitchToDefaultContent/>
<ClickElement locatorType="xpath" locatorExpression="//span[@class='wp-panel-button fonticon-expand-collapse-panel new-dashboard-menu-open-btn inactive']"/>
<CreateDashboard input="Test2"/>
<SwitchToDefaultContent/>
<OpenCompassApp quadrant="east" appname="3DPlay"/>
<ClickElement locatorType="xpath" locatorExpression="//span[@class='preview-icon fonticon-fonticon-pin']"/>
<SwitchToDefaultContent/>
<SwitchToStartWindow/>
<Logout/>
</TestCase>
```

*when user want to validate the thing that is not present in file then, user use "criteria=notexist" and "match=partial" and use want to delete the file after validating that in csv file when, user have to use "delete=true" as shown in the below*

- **example.**

```
<ReadXML unziplocation="C:\Users\SGSPC022\Downloads\CreateDashboard\" unzipped="true" id="downloadedFile" fromBrowserDownloadLocation="true" regexExpr="true" downloadFileExt="zip" deleteFile="false"/>

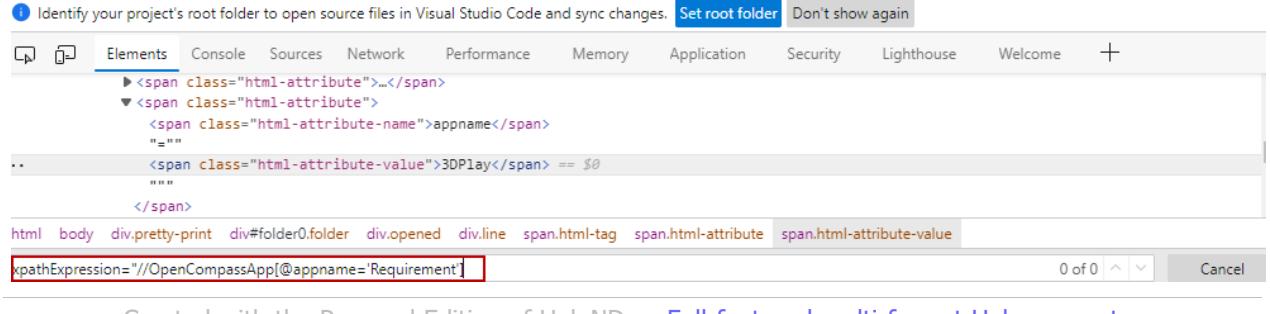
<AssertXML refid="downloadedFile" criteria="notexist" match="Partial xpathExpression="//OpenCompassApp[@appname='Requirement']" regexExpr="true" deleteFile="true" />
```

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```

<TestCase xmlns="https://www.steepgraph.com" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="https://www.steepgraph.com ../../../../../resources/xsd/TestAutomationFramework.xsd">
 <Login username="$csv{username}" password="$csv{password}" />
 <SwitchToDefaultContent/>
 <setStartWindow/>
 <OpenCompassApp quadrant="east" appname="3DDashboard"/>
 <SwitchToWindow/>
 <Wait time="1000"/>
 <SwitchToDefaultContent/>
 <ClickElement locatorType="xpath" locatorExpression="//span[@class='wp-panel-button fonticon fonticon-expand-collapse-panel new-dashboards-menu-open-btn']"/>
 <ClickElement locatorType="xpath" locatorExpression="//span[@class='wp-panel-button fonticon fonticon-menu new-dashboard-menu-open-btn']"/>
 <CreateDashboard input="Test1"/>
 <SwitchToDefaultContent/>
 <OpenCompassApp quadrant="east" appname="3DPlay"/>
 <!-- <ClickElement locatorType="xpath" locatorExpression="//span[@class='preview-icon fonticon fonticon-pin']"/> -->
 <ClickElement locatorType="xpath" locatorExpression="//span[@class='pin-icon preview-icon ifwe-action-icon fonticon fonticon-pin click-area']"/>
 <ClickElement locatorType="xpath" locatorExpression="//button[@class='btn-primary btn btn-root']"/>
 <SwitchToDefaultContent/>
 <SwitchToStartWindow/>
 <Logout/>
</TestCase>

```



Created with the Personal Edition of HelpNDoc: [Full-featured multi-format Help generator](#)

## ApplySearch

### ApplySearch:

User can use applySeach tag when use choosing advance search type then this tag will used for applying the Search filter form.

- Syntax:

```
<ApplySearch/>
```

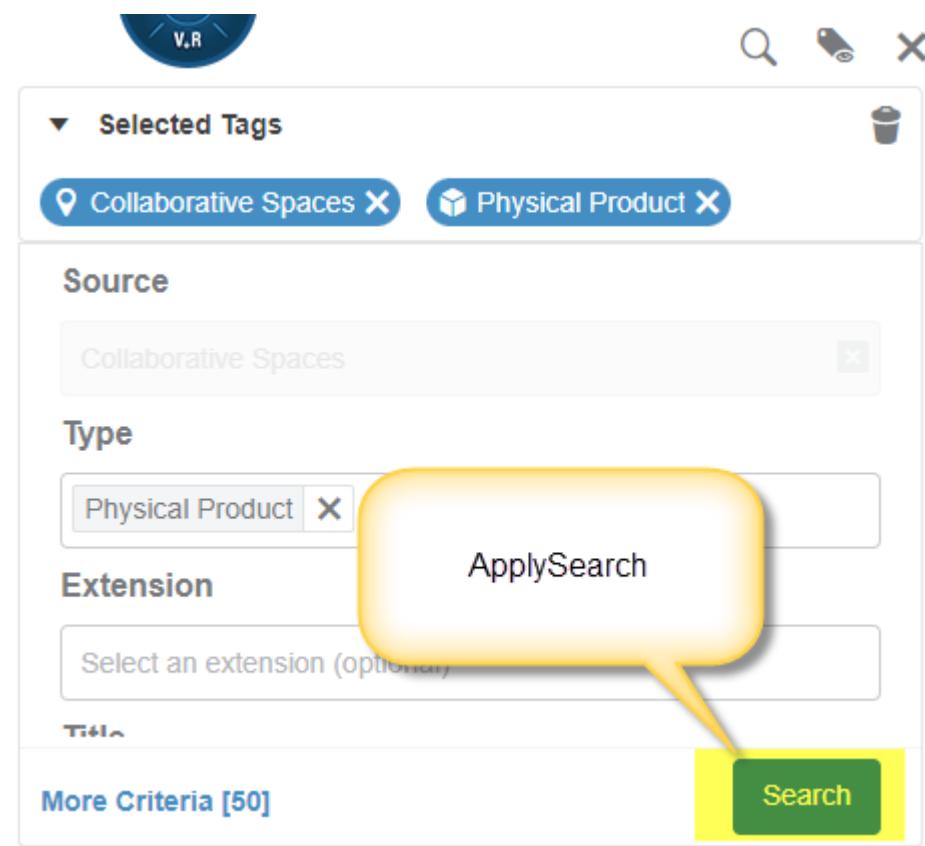
- Example:

```

<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<TestCase xmlns="https://www.steepgraph.com" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="https://www.steepgraph.com ../../../../../resources/xsd/TestAutomationFramework.xsd">

 <Login username="$csv{Username}" password="$csv{Password}" />
 <ClickElement locatorExpression="//div[@id='searchFieldDropdown']" locatorType="xpath"/>
 <ClickElement locatorExpression="//li[@title='Advanced Search in the GitTagger']" locatorType="xpath"/>
 <Wait time="5000"/>
 <InputText locatorType="xpath" locatorExpression="//input[@placeholder='Select a type (optional)']" input="Physical product"/>
 <ClickElement locatorType="xpath" locatorExpression="//span[text()='Physical Product']" />
 <ApplySearch/>
 <Wait time="5000"/>
<Logout/>
</TestCase>

```



Created with the Personal Edition of HelpNDoc: [Say Goodbye to Documentation Headaches with a Help Authoring Tool](#)

## CallWebService

### CallWebService:

User can use CallWebService tag to perform web services related operations.

#### ■ Syntax:

```
<CallWebService method="get" url="" id="" comparewith="" type="REST"
criteria="contains" key="" />
```

#### ■ Attribute:

- **id:(mandatory)**

Reference for storing response of webservice.

- **method:(mandatory)**

There are four methods for CallWebService which are as follows:

1. **Get:** GET method is used to retrieve data from a server at the specified resource.
2. **Post:** POST requests are used to send data to the API sever to create or update a resource. The data sent to the server is stored in the request body of the HTTP request.
3. **Delete:** The DELETE method is exactly as it sounds: delete the resource at the specified URL.
4. **Put:** Similar to POST, PUT requests are used to send data to the API to create or update a resource. The difference is that PUT requests are idempotent. That is, calling the same PUT request multiple times will always produce the same result. In contrast, calling a POST request repeatedly make have side effects of

creating the same resource multiple times.

- **url:(mandatory)** Url of webservice
- **comparewith:(mandatory)** Part of key value or whole value for comparison.
- **type:(mandatory)**

There two types of services which are as follows:

  1. **SOAP:** SOAP is known as the Simple Object Access Protocol. SOAP was developed as an intermediate language so that applications built on various programming languages could talk quickly to each other and avoid the extreme development effort.
  2. **REST:** REST stands for Representational State Transfer. REST is used to build Web services that are lightweight, maintainable, and scale-able.
- **criteria:(mandatory)**

There are five conditions user can apply for webservice tag which are as follows:

  - 1.Contains
  - Equals to
  - Not equals to
  - Greater than
  - Less Than
- **key: (mandatory)**  
Key element which user wants to compare.
- **errormessage:(optional)** User can give hardcoded message in this attribute.
- **username:(optional)** This attribute provides username for login.
- **password: (optional)** This attribute provides password for login.
- **soapaction :(optional)**
- **basicAuth:(optional)**
- **refid: (optional)** id attribute of FindElement tag will used as refid for this tag.

#### ▪ Example:

If user wants to get response of any webservices in that case, following tag should be used.

```
<CallWebService method="get"
url="https://jsonplaceholder.typicode.com/todos/1" id="Test"
comparewith="delectus" type="REST" criteria="contains" key="title"/>
```

---

Created with the Personal Edition of HelpNDoc: [Free EBook and documentation generator](#)

## CheckElement

### CheckElement :

Check if an element satisfies, specified criteria for further processing using provided locatorType and locatorExpression.

#### ▪ Syntax:

```
<CheckElement locatorType="" locatorExpression="" criteria="" />
```

▪ **Attributes:**

- **locatorExpression:(mandatory)**

This attribute contains expression specific to locator type

- **locatorType:(mandatory)**

There are 4 types of locator: id, cssselector, name, and xpath which defines locator type.

- **wait :(optional)**

This attribute contains wait time in milliseconds to find element. by default '3dx-tas.execution.step.timeout'.

- **criteria:(mandatory)**

Specifies the criteria on which the element will be tested.

The possible values for this attribute are:

1. visible
2. invisible
3. editable
4. noneditable
5. disabled
6. enabled
7. class: disabled

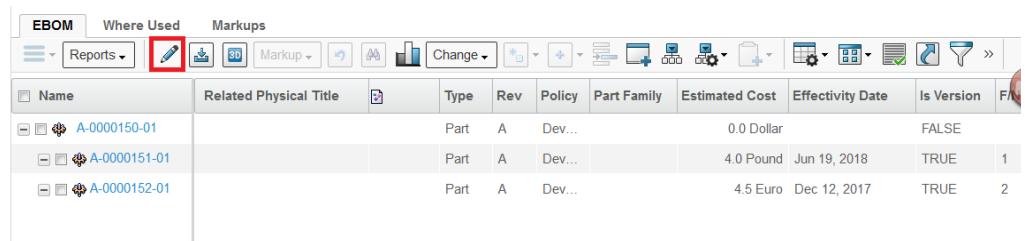
**NOTE:**

1. visible, invisible ==> works on any html element. It checks that element displayed or not
2. editable, noneditable ==> works on editable form. In view mode it can be used.
3. enable, disabled ==> works on any html to check, that element is enabled or disabled.
4. class: disabled ==> works on any html to check that class of that element disabled, so the element is disabled

▪ **Examples:**

In the images below, the user wants to check the visibility of the edit link on EBOM page. In this case the tag will be:

```
<CheckElement locatorType="cssselector"
locatorExpression="td[id='ENCIndentedBOMEEditAll']>img" criteria="visible"/>
```



Name	Related Physical Title	Type	Rev	Policy	Part Family	Estimated Cost	Effectivity Date	Is Version	F
A-0000150-01		Part	A	Dev...		0.0 Dollar		FALSE	
A-0000151-01		Part	A	Dev...		4.0 Pound	Jun 19, 2018	TRUE	1
A-0000152-01		Part	A	Dev...		4.5 Euro	Dec 12, 2017	TRUE	2

The edit link is visible on the page. Hence the check passes and further processing on the link can be done.

**Examples:**

In the images below, the user wants to check the criteria is editable on the

selected element,in this case the tag will be:

```
<CheckElement locatorType="id" locatorExpression="Name"
criteria="editable" />
```

Create Part

Fields in red italics are required	
Type	Part
Name	<input style="background-color: #ffff00;" type="text"/>
AutoName Series	General Series
Title	<input type="text"/>
Description	<input type="text"/>
Phase	Development
Options	<input type="checkbox"/> Enable Change Required <input type="checkbox"/> Configured <input type="checkbox"/> Design Collaboration

### Examples:

In the images below,the user wants to check the criteria is enabled on the selected element,in this case the tag will be:

```
<CheckElement locatorType="id" locatorExpression="VPMProductName1"
criteria="enabled" />
```

Create Part

Fields in red italics are required	
Type	Part
Name	<input type="text"/>
AutoName Series	General Series
Title	<input style="background-color: #ffff00;" type="text"/>
Description	<input type="text"/>

In simple terms: **Enable = To turn On. Disable = To Turn Off.**( no dropdown contains)

When you turn on a feature you enable it.

When you turn a feature off you disable it.

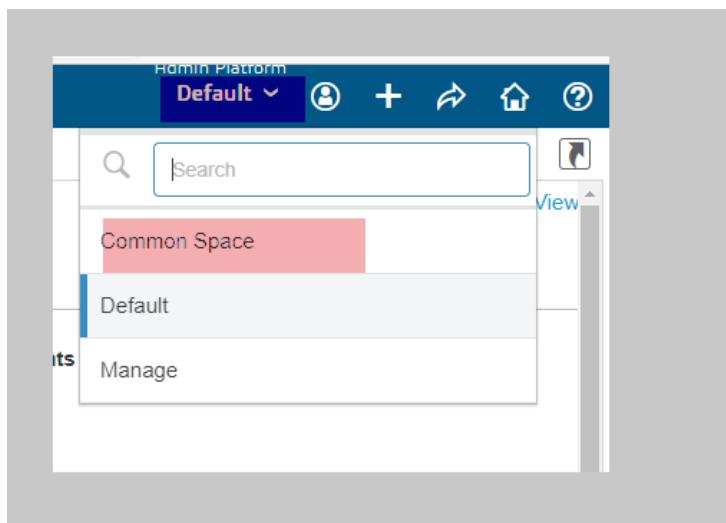
**Examples:**

In the images below, the user wants to check the criteria is noneditable on the selected element, in this case the tag will be:

```
<CheckElement locatorType="xpath"
locatorExpression="//input[@id='Project_txt']" criteria="noneditable"/>
```

In the images below, the user wants to check the criteria is disabled on the selected element, in this case the tag will be:

```
<CheckElement locatorType="xpath"
locatorExpression="//input[@id='Project_txt']" criteria="disabled"/>
```



A screenshot of a modal dialog box titled 'Credentials'. The dialog contains several input fields and sections:

- Credentials**: A section with a lock icon.
- Collaborative Space**: A section with a dropdown menu containing 'Common Space' (highlighted with a yellow background).
- Organization**: A section with a 'Company Name' input field.
- Access Role**: A section with a dropdown menu containing 'Author'.

At the bottom of the dialog are 'OK' and 'Cancel' buttons.

## CheckIndentedTableRow

### CheckIndentedTableRow:

Check if a cell in a selected row in an indented table satisfies specified criteria for further processing using provided locatorType and locatorExpression.

#### Note:

The row in which the cell to be tested is located must be selected using SelectIndentedTableRow tag to use this tag.

#### ▪ Syntax:

```
<CheckIndentedTableRow position="" refid="" criteria="" columnlabel="" />
```

#### ▪ Attributes:

- **position:(mandatory)** Specifies the column number in the indented table found out by inspecting the DOM.
- **refid:(mandatory)** Refers to the id attribute of SelectIndentedTableRow tag.
- **criteria:(mandatory)** Specifies the criteria on which the element will be tested.

The possible values for this attribute are:\

1. visible
2. invisible
3. editable
4. noneditable

- **columnlabel:(mandatory)**

Specifies the label of the indented table column.

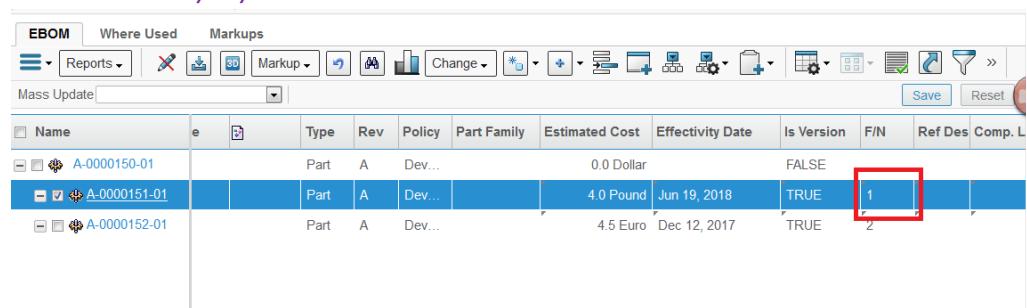
- **insideFreezePane:(optional)**

Possible values are 'true' or 'false'. If criteria is true it will check inside the freeze panel.

#### ▪ Examples:

In the images below, the user wants to check the editability of the F/N cell on the EBOM page. Its position is 11 as found by inspecting DOM. In this case the tag will be:

```
<CheckIndentedTableRow position="11"
refid="id_attribute_in_SelecteIndentedTablerow" criteria="editable"
columnlabel="F/N"/>
```



The screenshot shows the SAP ERP interface with the EBOM (Engineering Bill of Materials) module open. The top navigation bar includes tabs for 'EBOM', 'Where Used', and 'Markups'. Below the navigation bar is a toolbar with various icons for document management and search. The main area displays a table with several columns: Name, e, Type, Rev, Policy, Part Family, Estimated Cost, Effectivity Date, Is Version, F/N, Ref Des, and Comp. L. There are three rows of data. The first row contains the part number A-0000150-01. The second row contains A-0000151-01, with the F/N column value highlighted by a red box. The third row contains A-0000152-01. The F/N column is the 11th column in the table.

Name	e	Type	Rev	Policy	Part Family	Estimated Cost	Effectivity Date	Is Version	F/N	Ref Des	Comp. L
A-0000150-01		Part	A	Dev...		0.0 Dollar		FALSE			
A-0000151-01		Part	A	Dev...		4.0 Pound	Jun 19, 2018	TRUE	1		
A-0000152-01		Part	A	Dev...		4.5 Euro	Dec 12, 2017	TRUE	2		

The cell is editable in this case. Hence the test will pass and further processing on it can be done.

Created with the Personal Edition of HelpNDoc: [Produce electronic books easily](#)

## ClickBackButton

### ClickBackButton:

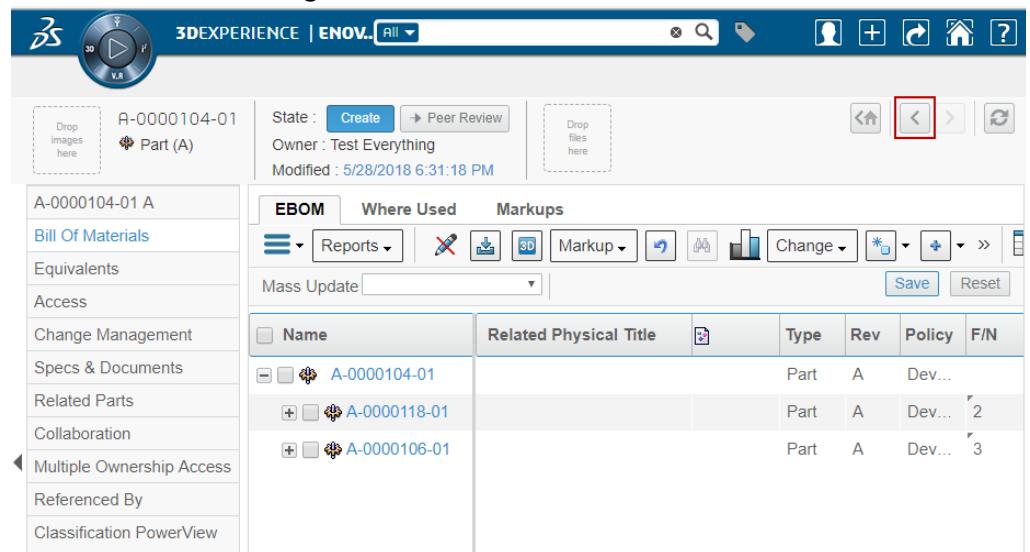
Clicks the back button on the page header.

- **Syntax:**

*<ClickBackButton/>*

- **Examples:**

Please check below image.



Created with the Personal Edition of HelpNDoc: [Create cross-platform Qt Help files](#)

## ClickCategoryCommand

### ClickCategoryCommand:

Click category menu command.

- **Syntax:**

*<ClickCategoryCommand title=" "/>*

▪ **Attributes:**

- **title:(mandatory)**

The title attribute is used to provide the name of the category menu command that the user wants to be clicked.

▪ **Examples:**

In the image below the user wants the script to click the category menu command “Bill Of Materials” for type Part, then the tag will be as follows

`<ClickCategoryCommand title="Bill Of Materials"/>`




---

Created with the Personal Edition of HelpNDoc: [Free Web Help generator](#)

---

## ClickDashboardCheckbox

### ClickDashboardCheckbox:

This tag is to select and unselect objects using check-boxes in 3ddashboard.

▪ **Syntax:**

`<ClickDashboardCheckbox action="" rows="" />`

▪ **Attributes:**

- **action:(mandatory)**

This attribute has possible values as check, unCheck, checkAll and unCheckAll.

- **rows: (optional)**

Provide the row number which needs to be checked or unchecked. In case of multiple rows, provide pipe separated values.

▪ **Examples:**

`<ClickDashboardCheckbox action="check" rows="1/3/5" />`

This tag is use when you are in 3DDashboard from there only you can select row as shown in picture.

	Display Name	Revision	Type	Covers	Refined Into	Parameter Value	State
1	Requirement00000011 A	A	Requirement				In Work
2	Requirement00000013 A	A	Requirement				In Work
3	Requirement00000014 A	A	Requirement				In Work
4	Requirement00000015 A	A	Requirement				In Work
5	Requirement00000016 A	A	Requirement				In Work
6	Requirement00000017 A	A	Requirement				In Work

Created with the Personal Edition of HelpNDoc: [Full-featured Help generator](#)

## ClickElement

### ClickElement:

Clicks Web Element to be identified, using given locatorType and LocatorExpression.

- **Syntax:**

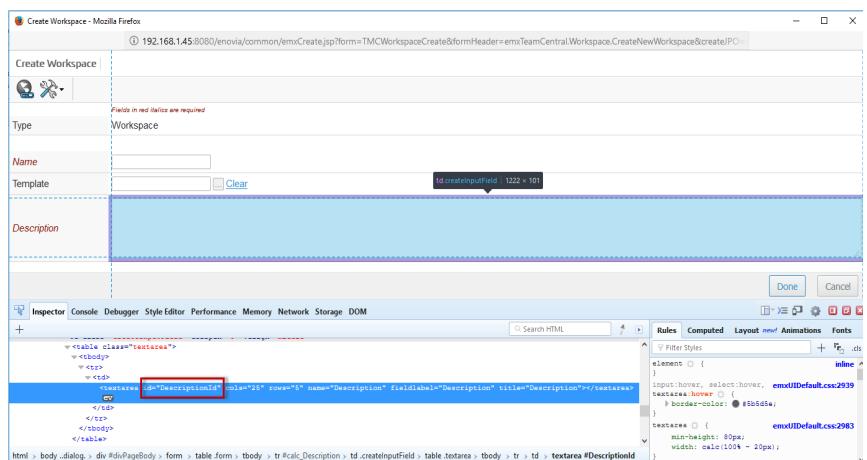
```
<ClickElement locatorType="" locatorExpression="" />
```

- **Attributes:**

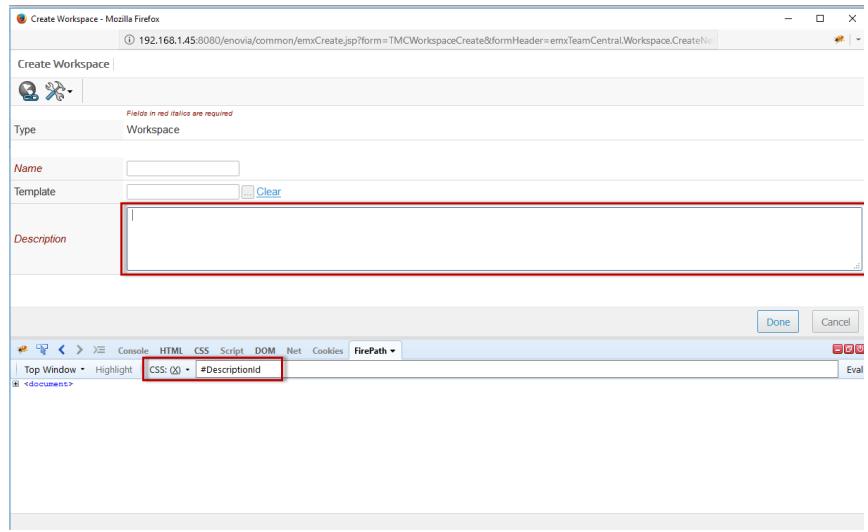
- **locatorType:(mandatory)**

There are 4 types of locator as mentioned below:

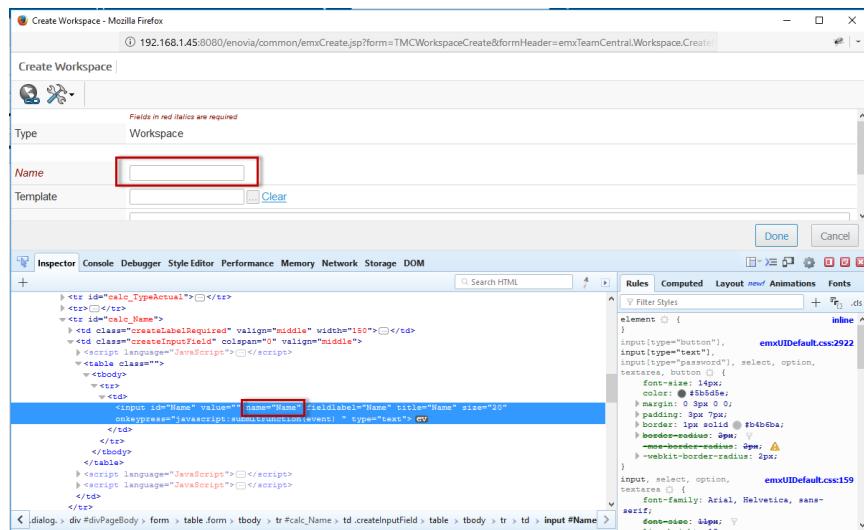
1. **id:** Find the element on web page using id attribute of web element. Please check below image:



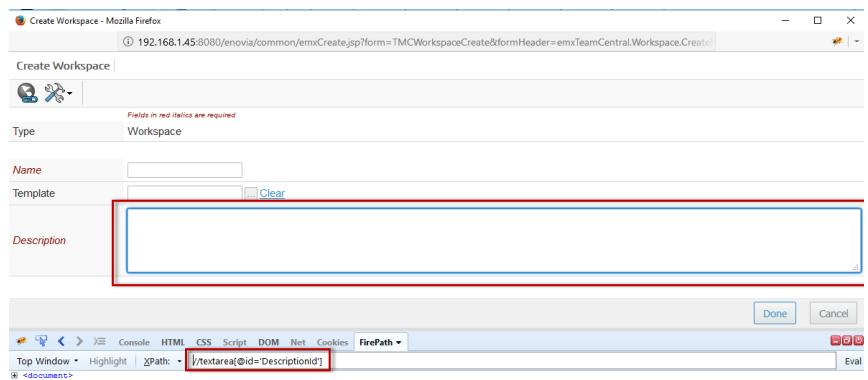
2. **cssSelector:** Find element on web page using css expression. Please check below image:



**3. name:** Find element on web page using name attribute of web element expression. Please check below image:



**4. xpath:** Find element on web page using xpath expression. Please check below image:



Use fire path to test xpath on browser before using it on test case xml. [Firepath](#) is add-in available for Firefox.

- **locatorExpression:(mandatory)**

This attribute contains expression specific to locator type. If locatorType is id then locatorExpression contain id of the web element, or if locatorType is xpath, then xpath expression should be given in this attribute.

- **highlight:(optional)**

This attribute is used to choose whether to highlight the element when working on it. Takes values true or false. Default value is false.

- **mode:(optional)**

This attribute is used to select execution action, viz. selenium/js based. Takes values selenium or js. Default value is selenium.

- **async:(optional)**

The possible values are 'true' or 'false'. Default value is false.

- **style:(optional)**

This attribute is used to provide styling to the highlight used. Default value is taken from 3DXtAS Property key '`3dx-tas.highlight.webelement.style`'. Only if highlight is true.

- **retryCount:(optional)**

This attribute will retry click if the element is not selected after click till the element is clicked or for given retryCount (default 2).

- **criteria:(optional)**

This attribute validates click event for further conditions based on criteria. Possible values are :

- selected : checks if element is selected after click
- unselected : checks if element is not selected after click
- invisible : checks if element is not visible after click

- **Examples:**

```
<ClickElement locatorType="id" locatorExpression="DescriptionId"/>
<ClickElement locatorType="xpath"
locatorExpression="//textarea[@id='DescriptionId']"/>
<ClickElement locatorType="xpath" locatorExpression="//input[@id='rbmbrow-0']"
criteria="selected" retryCount="3"/>
```

---

Created with the Personal Edition of HelpNDoc: [Streamline your documentation process with HelpNDoc's WinHelp HLP to CHM conversion feature](#)

## [ClickForwardButton](#)

### **ClickForwardButton:**

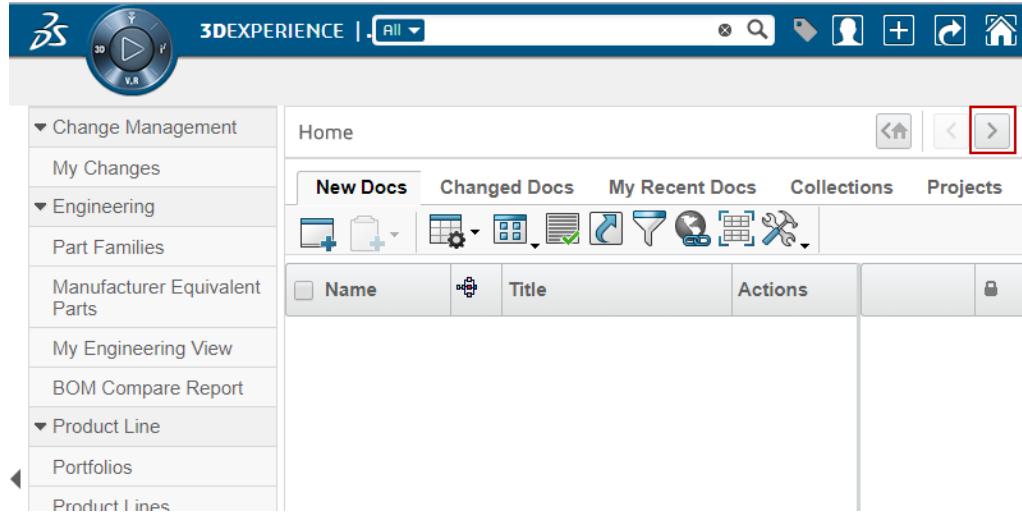
Clicks the forward button on the page header.

- **Syntax:**

*<ClickForwardButton/>*

▪ **Examples:**

Please check below image.



Created with the Personal Edition of HelpNDoc: Leave the tedious WinHelp HLP to CHM conversion process behind with HelpNDoc

## ClickGlobalActionsMenu

### ClickGlobalActionsMenu:

Click given global action menu/command.

▪ **Attributes:**

- **highlight:(optional)**

This attribute used is choose whether to highlight the element when working on it. Takes values true or false.

- **style:(optional)**

This attribute is used to provide styling to the highlight used. Default value is taken from 3DXtAS Property key '`3dx-tas.highlight.webelement.style`'. Only if highlight is true.

- **CommandLabel:(Mandatory)**

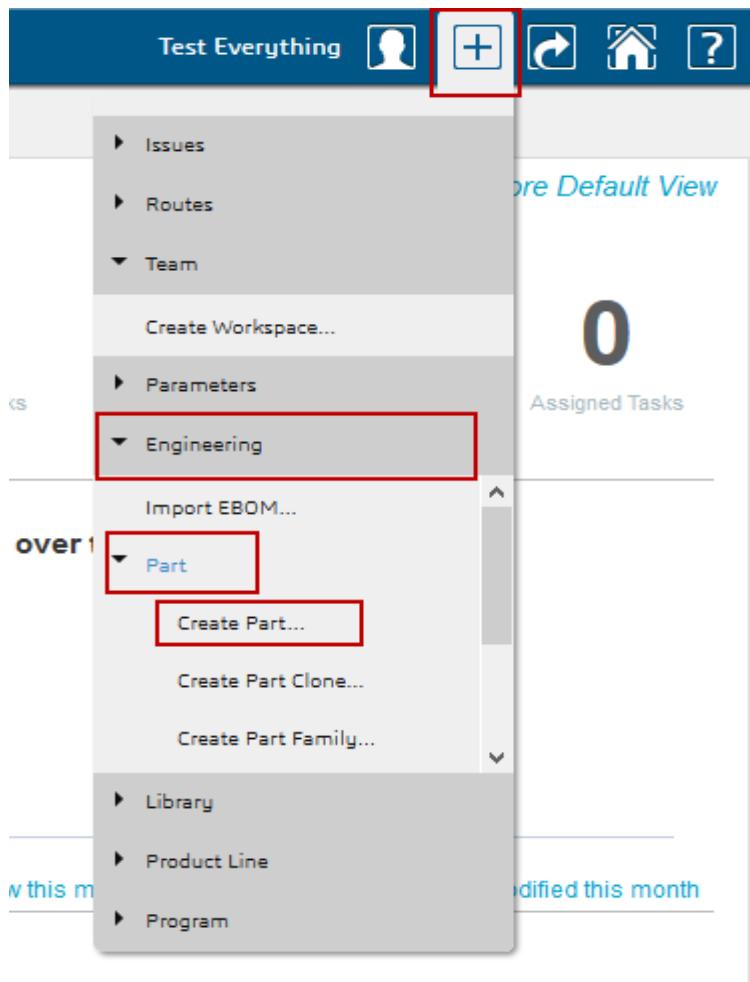
User must pass command label. If command to execute is inside another command, then user need to pass parent menu/command label and child command label separated by pipe (|).

▪ **Examples:**

`commandLabel = 'Engineering|Part|Create Part...'`

▪ **Syntax:**

*<ClickGlobalActionsMenu commandLabel=""/>*



Created with the Personal Edition of HelpNDoc: [Modernize your help files with HelpNDoc's WinHelp HLP to CHM conversion tool](#)

## ClickGlobalToolsMenu

### ClickGlobalToolsMenu:

Click given global tools menu/command.

#### ▪ Attributes:

- **commandLabel:(Mandatory)**

User has to pass command label. If command to execute is inside another command, then user need to pass parent menu/command label and child command label separated by pipe (|).

- **highlight:(optional)**

Possible values are true or false, if want to highlight the action keep highlight attribute as true else false.

- **style:(optional)**

This attribute is used to provide styling to the highlight used. Default value is taken from 3DXtAS Property key '`3dx-tas.highlight.webelement.style`'. Only if highlight is true.

#### ▪ Syntax:

```
<ClickGlobalToolsMenu commandLabel="" />
```

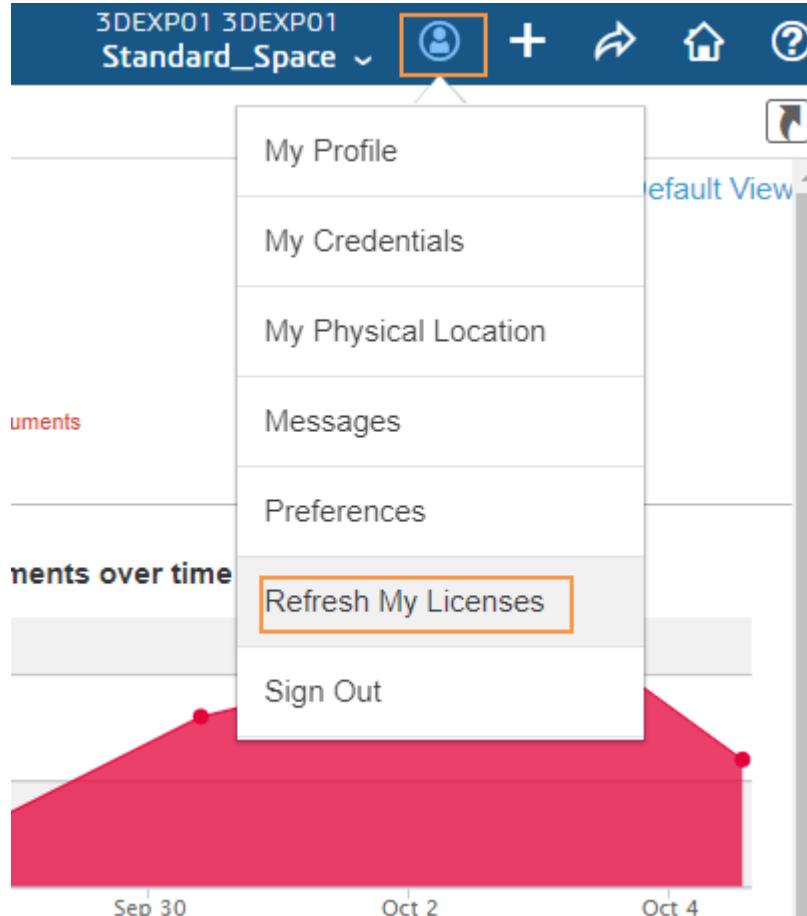
User must pass command label. If command to execute is inside another command, then user need to pass parent menu/command label and child

command label separated by pipe (|).

- **Examples:**

```
<ClickGlobalToolsMenu commandLabel="Profile | Refresh My Licenses"
highlight="true" style="Calibri Light (Headings)" />
```

To Click given global tools menu/command this can be used. With the help of highlight attribute that element can highlighted.




---

Created with the Personal Edition of HelpNDoc: [Free PDF documentation generator](#)

---

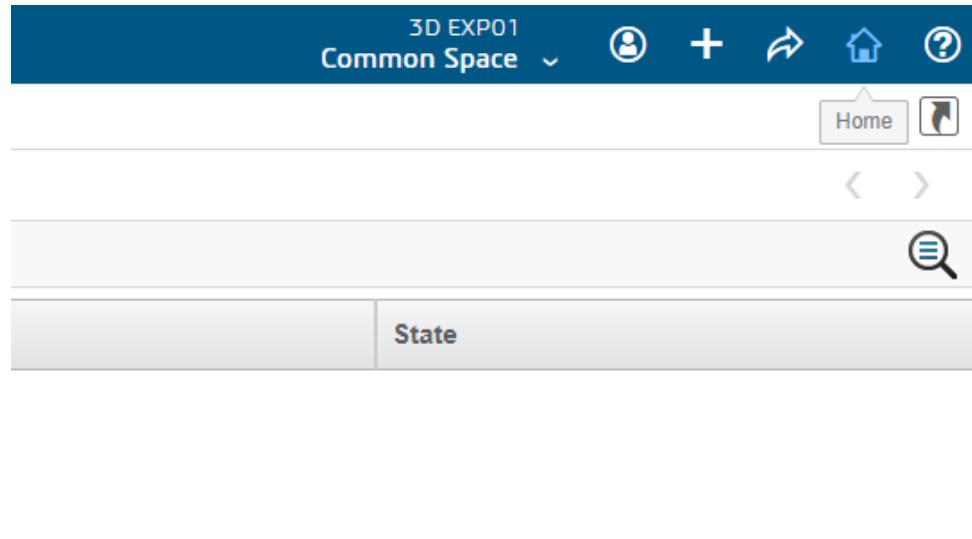
## **ClickHomeMenu**

### **ClickHomeMenu:**

User can use ClickHomeMenu to click Home on Top Bar

- **Syntax:**

```
<ClickHomeMenu />
```



Created with the Personal Edition of HelpNDoc: [Easy to use tool to create HTML Help files and Help web sites](#)

## ClickMyDeskMenu

### ClickMyDeskMenu:

Click given my desk menu/command.

- **Attributes:**

- **highlight:(optional)**

Possible values are true or false, if want to highlight the action keep highlight attribute as true else false.

- **style:(optional)**

This attribute is used to provide styling to the highlight used. Default value is taken from 3DXtAS Property key '`3dx-tas.highlight.webelement.style`'. Only if highlight is true.

- **CommandLabel:(Mandatory)**

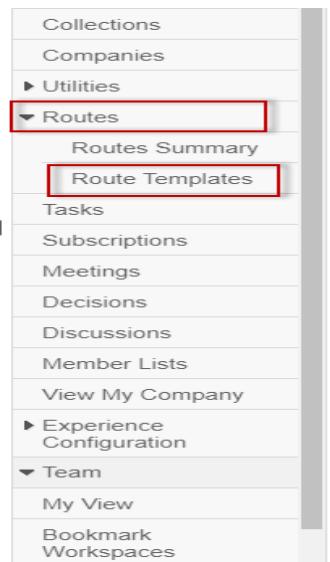
User must pass command label. If command to execute is inside another command, then user need to pass parent menu/command label and child command label separated by pipe (|).

- **Examples:**

`commandLabel = 'Routes|Route Templates...'`

- **Syntax:**

`<ClickMyDeskMenu commandLabel="">`




---

Created with the Personal Edition of HelpNDoc: [Easy Qt Help documentation editor](#)

---

## ClickOverrideLink

### ClickOverrideLink:

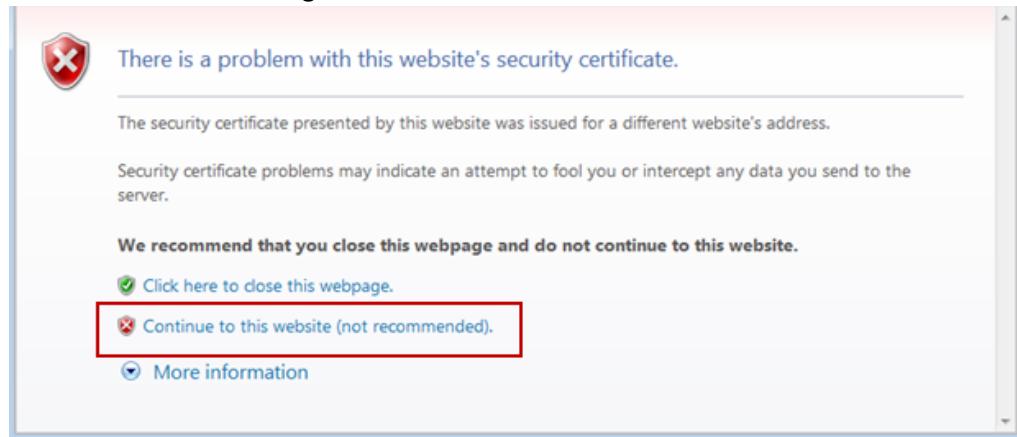
When there is no valid certificate available for https connection, browser will show this link(certificate error) while opening new window. To proceed further, script need to click on this link.

This tag is now **Deprecated**, Please refer `Browser.property settings 3dx-tas.browser.acceptInsecureCerts=true`

- **Syntax:**

*<ClickOverrideLink/>*

Please check below image:




---

Created with the Personal Edition of HelpNDoc: [iPhone web sites made easy](#)

---

## ClickPortalCommand

### ClickPortalCommand:

Switch to given portal tab.

- **Attributes:**

- **title:(Mandatory)**

In title attribute user must specify portal command title to open.

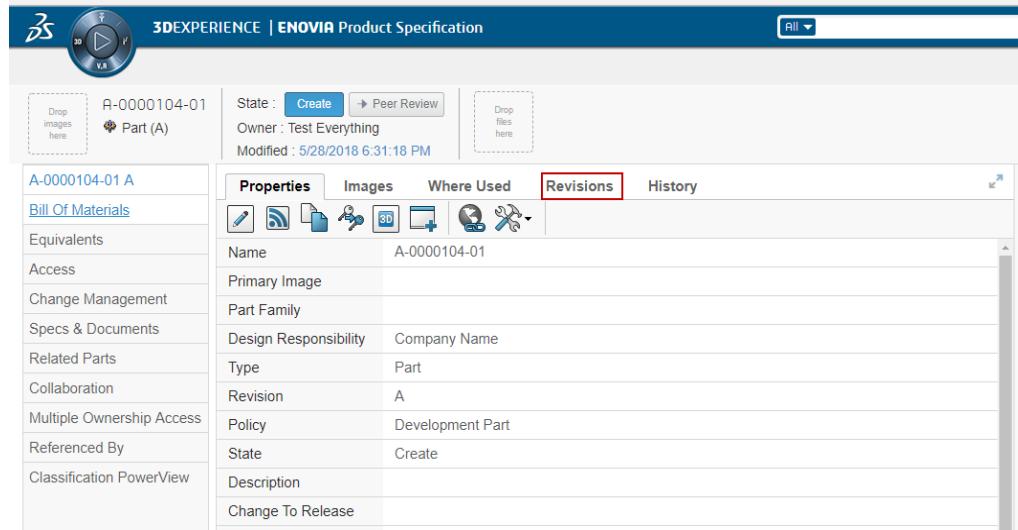
- **Syntax:**

```
<ClickPortalCommand title="" />
```

- **Examples:**

Please check below image. Suppose we want to open revisions tab, then we need to define this tag like:

```
<ClickPortalCommand title="Revisions" />
```



Before using this tag, make sure that you have switch to frame which contains this portal.

Created with the Personal Edition of HelpNDoc: [Transform Your Word Document into a Professional eBook with HelpNDoc](#)

## ClickRefreshButton

### ClickRefreshButton:

Clicks the refresh button on the page header.

- **Attributes:**

- **target:(optional)**

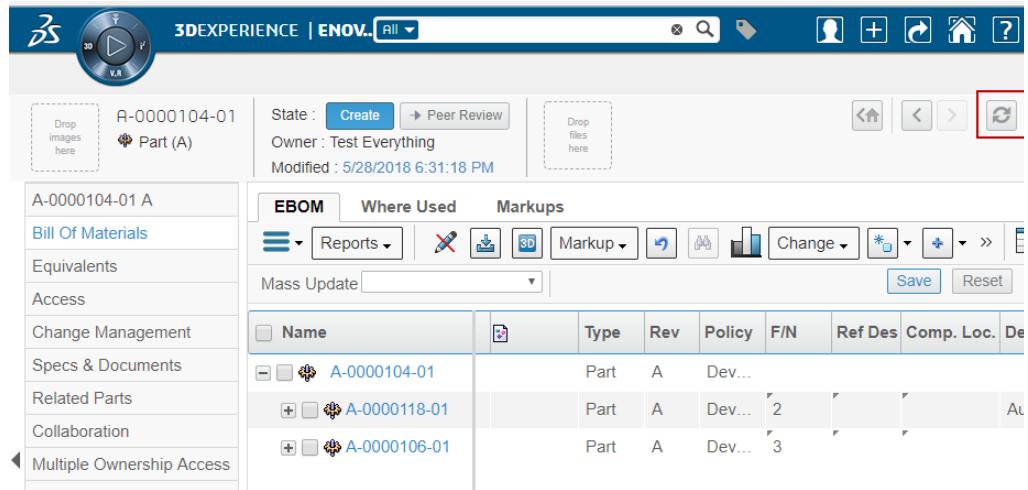
possible values are header for widget refresh button without pin to dashboard & another value is tab for refresh the tab after minimise the widget

- **Syntax:**

```
<ClickRefreshButton />
<ClickRefreshButton target="tab" />
<ClickRefreshButton target="header" />
```

- **Example:**

Please check below image.



Created with the Personal Edition of HelpNDoc: [Save time and frustration with HelpNDoc's WinHelp HLP to CHM conversion feature](#)

## ClickWindowElement

### ClickWindowElement:

Click any element on the browser screen using an image of the element.

- **Syntax:**

```
<ClickWindowElement imagepath="" />
```

- **Attributes:**

- **imagepath:(mandatory)**

Specifies the path where the image of the screen element is located.

- **skiperror:(optional)**

Specifies whether to skip current test case or not. Values are true or false. Should be set to false to run test case. Default value is false.

- **refid:(optional)**

The id attribute of FindElement tag will be used as refid for this tag.

- **wait:(optional)**

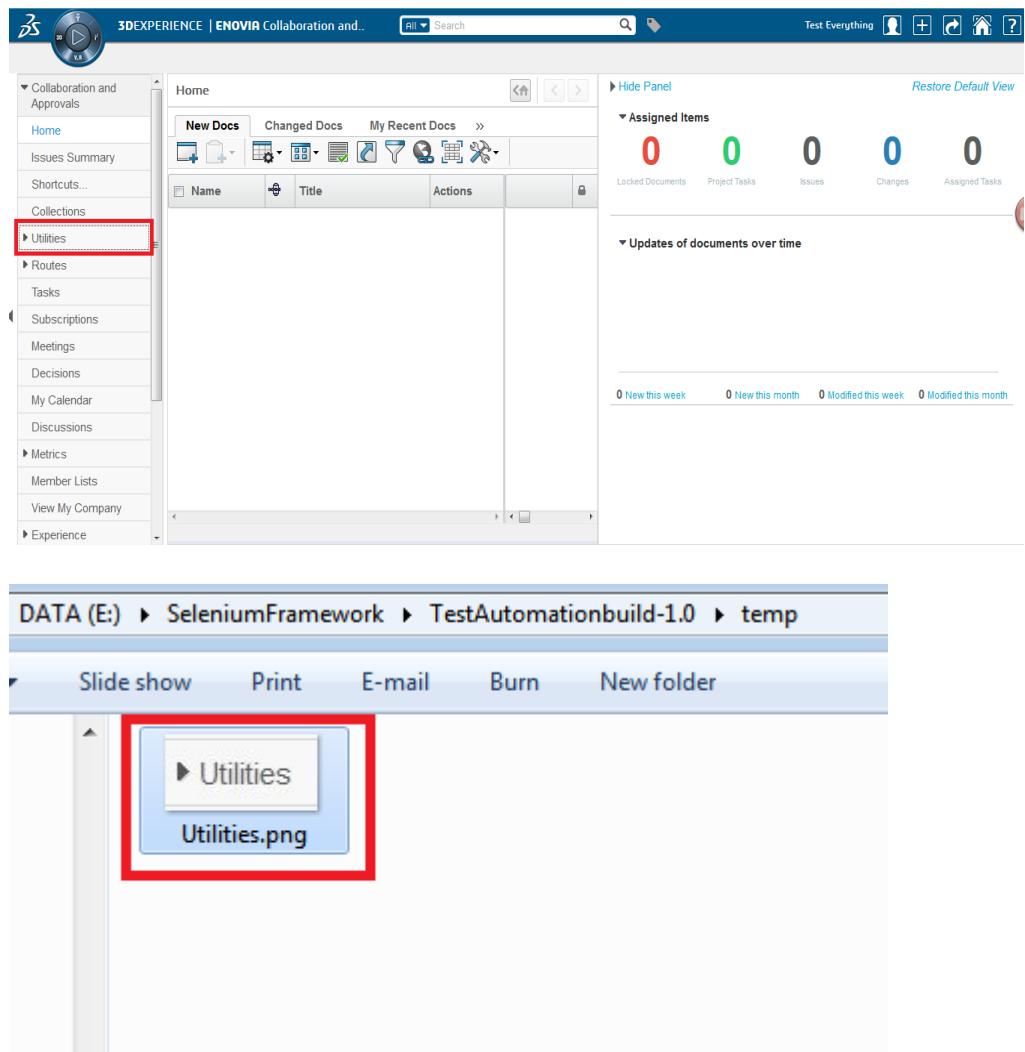
This attribute contains wait time in MS to find element. default wait will be taken from 3DX\_TAS propertie's "3dx-tas.execution.step.timeout" key. Preferred wait is 10000.

- **Examples:**

In the image below, the user wants to have a menu called Utilities clicked. The tag in this case will be as follows:

```
<ClickWindowElement imagepath="E:\SeleniumFramework\TestAutomationbuild-1.0\temp\Utilities.png" />
```

The image of the Utilities menu is located at the path E:\\SeleniumFramework\\TestAutomationbuild-1.0\\temp\\Utilities.png This will be used by the script to locate the menu on the screen and click it.



Created with the Personal Edition of HelpNDoc: [Don't be left in the past: convert your WinHelp HLP help files to CHM with HelpNDoc](#)

## CloseAllOtherWindow

### CloseAllOtherWindow:

Close all window except current window. This tag is used when user have open multiple window with help of this tag is close all other window and automatically move to currently window.

- **Attributes:**

- **dismissAlert:(optional)**

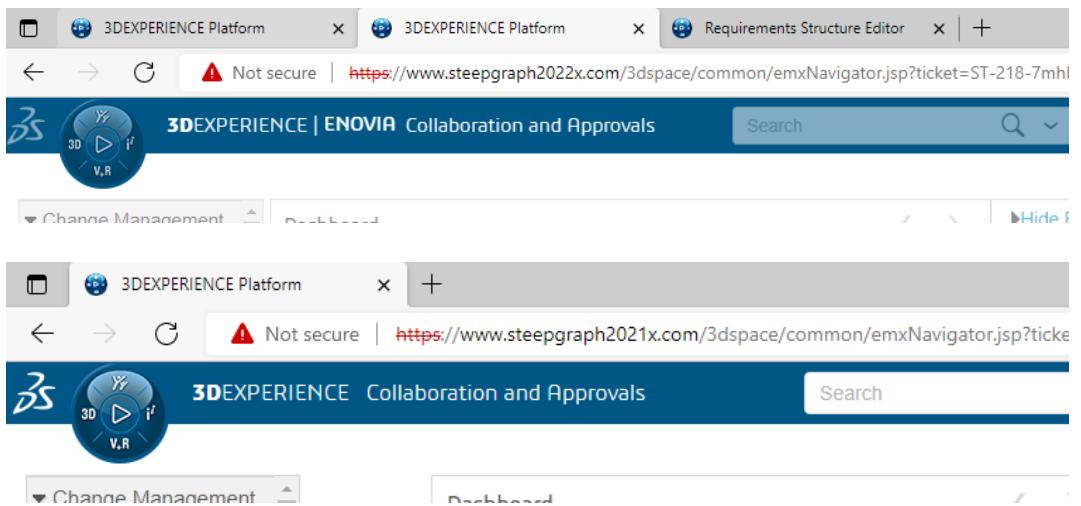
dismissAlert attribute is used to handle the alert while closing the all other window. Default value is false.

- **Syntax:**

`<CloseAllOtherWindow />`

- **Examples:**

As show in picture multiple tab is open. After using this tag is close all other tab and directly move to current window.



Created with the Personal Edition of HelpNDoc: [Write eBooks for the Kindle](#)

## **CloseCurrentWindow:**

### **CloseCurrentWindow:**

Close currently focused window.

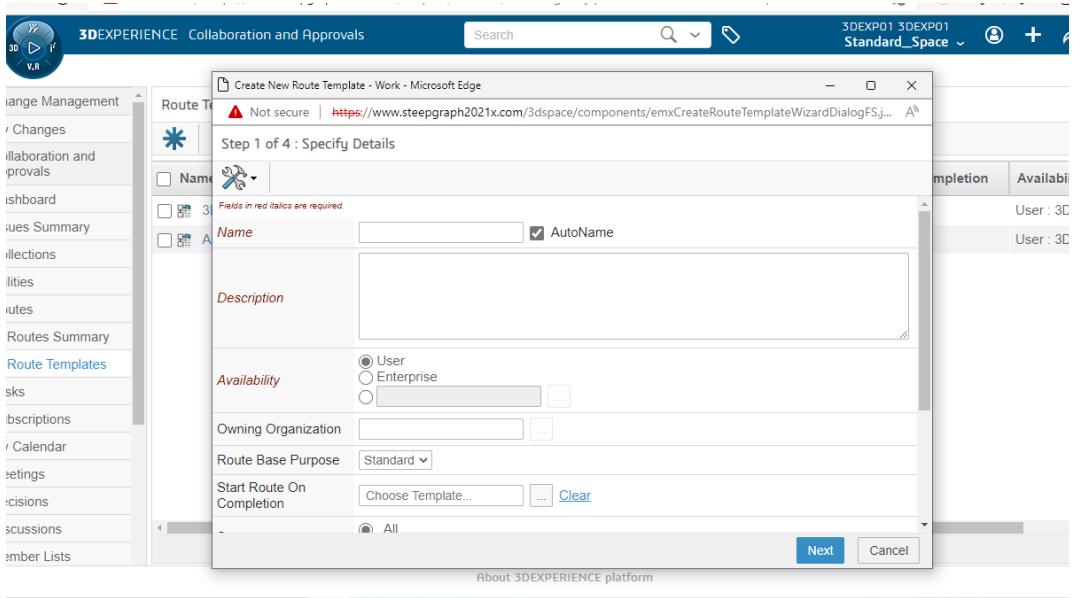
- **Syntax:**

```
<CloseCurrentWindow/>
```

User must use [SwitchToWindow](#) tag after this tag to move script control to required window.

- **Examples:**

As show in below picture when you want to closecurrentwindow, then this tag is used.



Created with the Personal Edition of HelpNDoc: [Maximize Your Documentation Capabilities with HelpNDoc's Project Analyzer](#)

## **CreateDashboard**

### **CreateDashboard :**

With the help of this tag, user can create a new dashboard with a certain name.

- **Syntax:**

```
<CreateDashboard input=" "/>
```

**NOTES:**

1. Using this tag user can Create a new Dashboard.
2. This tag takes an user input as input for Dashboard name.
3. Before creating new Dashboard, it will check whether any dashboard available or not with the same name. if yes then it will delete that dashboard & create a new one , else it will automatically create a Dashboard.
4. To Create the dashboard user have to present in 3DDashboard page.

- **Attributes:**

- **Input:(Mandatory)**

*This attribute takes userInput for naming the Dashboard.*

- **delete:(Optional)**

*this is used to delete the same named upper position dashboard if value is true.default value is false.*

- **wait:(Optional)**

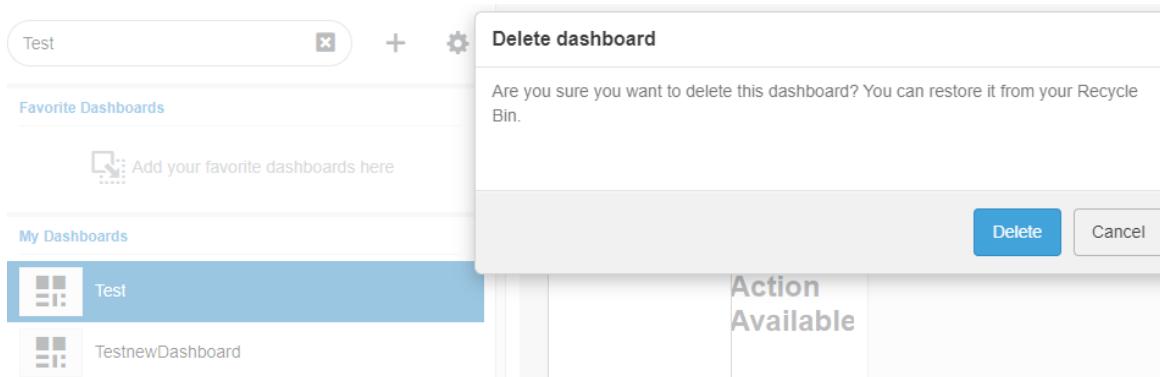
*This attribute contains wait time in milliseconds to find element.Default value will be taken from '3dx-tas.execution.step.timeout'.Prefered wait is 5000MS.*

- **Examples:**

Suppose you want to **CreateDashboard** it will search for that name first if that name is present than it **delete** that name and then **Create New Dashboard** that name.

This all thing happen with is **createDashboard** tag only. As shown in below image.

```
<CreateDashboard input ="Test" />
```



*It will First search of that name if it get than he will detect that dashboard and create new dashboard for that name as shown in picture.*

---

Created with the Personal Edition of HelpNDoc: [Easily create EBooks](#)

## CloseCurrentWindow

### CloseCurrentWindow:

this tag is use to close the current window.

- **Syntax:**

```
<CloseCurrentWindow />
```

---

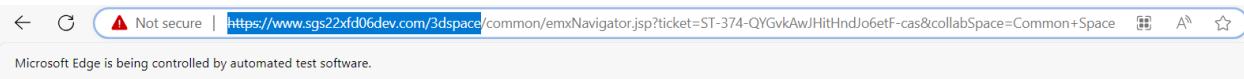
Created with the Personal Edition of HelpNDoc: [Free EPub producer](#)

## DeleteAllCookies

### DeleteAllCookies:

User can use DeleteAllCookies for Clearing cookies from users web browser. It will delete only the cookies related to current domain and current path.

For E.x If the execution window is given below. User is using DeleteAllCookies tag in this window.



Then this tag delete cookies related to this domain and path. It will delete this cookies shown in the below image.

Cookies	
CASTGC	
<b>JSESSIONID</b>	
<b>Name</b>	JSESSIONID
<b>Content</b>	E6075DDC9AAD45DD60110F747CE414C1
<b>Domain</b>	www.sgs22xfd06dev.com
<b>Path</b>	/3dspace
<b>Send for</b>	Secure same-site connections only

But it will not delete the cookies shown in the below image. As the path is different then the current domain.

**Cookies**

CASTGC

**Name**

CASTGC

**Content**

TGT-377-p4p9WTepd4qCblVrk269U1kGGuDlyxbTcot54xgJPKZL9bydf-cas

**Domain**

.sgs22xfd06dev.com

**Path**

/3dpassport/

**Send for**

Secure connections only

■ **Syntax:**

&lt;DeleteAllCookies/&gt;

■ **Example:**

As show in below example, when you **login** in to any site the **cookies** is created automatically for that site in browser.To delete that cookies this tags is use.

&lt;DeleteAllCookies/&gt;.

```
<ClickGlobalActionsMenu commandLabel="Product Line/Create Product Line" />
<SwitchToWindow />
<ClickElement locatorType="xpath" locatorExpression="//input[@name='autoNameCheck']" />
<InputText locatorType="id" locatorExpression="MarketingName" input="Market1"/>
<ClickElement locatorType="xpath" locatorExpression="//button[@class='btn-primary']" />
<SwitchToParentWindow />
<DeleteAllCookies/>
```

This are all cookies that generate automatically when you open or search for something.As shown in the below image.

The screenshot shows two adjacent pages from Microsoft Edge. The left page is the 'Settings' page, featuring a search bar at the top and a list of various settings categories. The right page is titled 'Cookies and data stored for sites you have browsed' and lists several entries with their respective sizes and cookie counts.

Site	Size	Cookies
3ds.com	0 bytes	0 cookies
microsoft.com	278 KB	0 cookies
sharepoint.com	21 bytes	3 cookies
steepgraph2021x.com	11 MB	1 cookie
steepgraph2022x.com	5 MB	1 cookie
steepgraphdev2022xdev.com	5 MB	1 cookie
steepgraphdev21x.com	5 MB	1 cookie

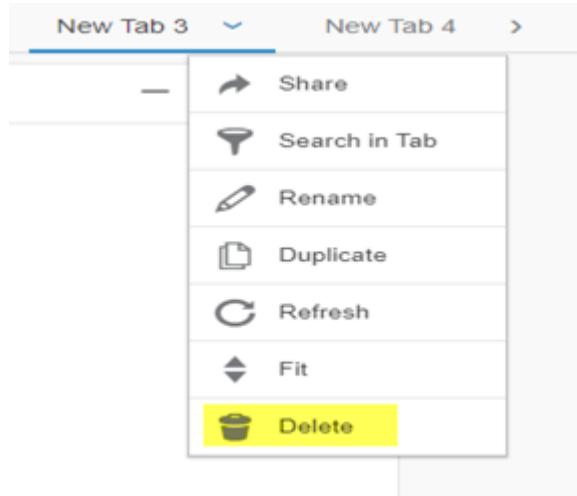
Created with the Personal Edition of HelpNDoc: [Full-featured EPub generator](#)

## DeleteCurrentTabInDashboard

### DeleteCurrentTabIn Dashboard:

This will delete currently controlled Tab in Dashboard shown in screenshot.

- **Syntax:**  
`<DeleteCurrentTabInDashboard />`
  
- **Attributes:**
  - **wait:(optional)**  
*This attribute contains wait time in milliseconds to find element. Default value will be taken from '3dx-tas.execution.step.timeout'. Preferred wait is 5000MS.*



Created with the Personal Edition of HelpNDoc: [Say Goodbye to Documentation Headaches with a Help Authoring Tool](#)

## DownloadFile

### DownloadFile(Deprecated TAG):

This tag is deprecated in TAS since 22xfd06, instead of this tag use **DownloadFileUsingCommand** or **DownloadFileUsingIcon** tag.

Download file using given [locatorType](#) and [LocatorExpression](#).

- **Syntax:**

```
<DownloadFile locatorType="" locatorExpression="" />
```

- **locatorExpression:(mandatory)**

This attribute contains expression specific to locator type

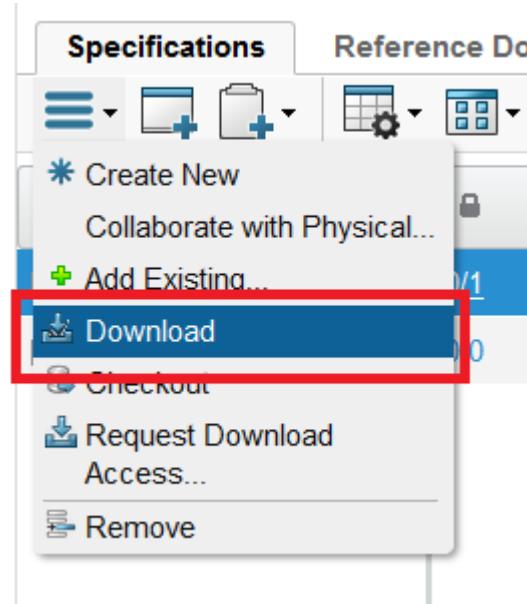
- **locatorType:(mandatory)**

There are 4 types of locator: id, cssselector, name, and xpath which defines locator type.

- **Example:**

In the image below the user has provided the locatorType and locatorExpression for the Download button in the DownloadFile tag. The script uses this to locate and click the Download button. So the tag will be:

```
<DownloadFile locatorType="xpath" locatorExpression="//label[text() = 'Download']"/>
```



Created with the Personal Edition of HelpNDoc: Write EPub books for the iPad

## DownloadFileUsingCommand

### DownloadFileUsingCommand:

Download the file using download command. commandlabel and (id/ input) is mandatory from 22xfd06 for validating the downloaded file.

If user don't want to check Download File then no need to provide id/input , it will automatically check the download folder specifies as 3dx-tas.download.filepath key in 3DX\_TAS.properties.

- Syntax:

```
<DownloadFileUsingCommand commandlabel="" />
```

- Attribute:

- **commandlabel:(mandatory)**

commandlabel attribute is the label of the download button in the action toolbar menu in UI.

- **wait: (optional)**

This attribute contains wait time in milliseconds to find element.Default value will be taken from '3dx-tas.execution.step.timeout'.Prefered wait will be based on the time it takes to download the whole file.

- **checkDownloadFile: (optional)**

this attribute is used for activating the download file validation, possible values are true , false. default value is false.

when checkDownloadFile value is true, then input/id should be used in the tag to give the location for the file Assertion.

- **input: (optional)**

this attribute is used to provide a hard-coded value for input. value of the downloaded filepath which includes the file should pass as input.

- **id: (optional)**

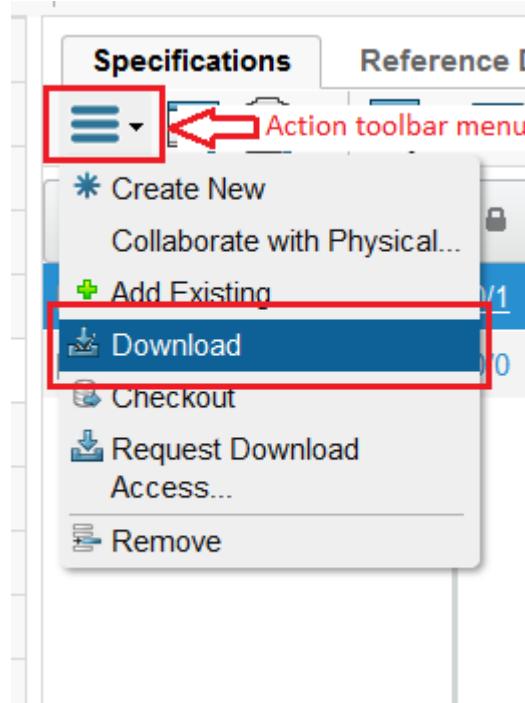
id(CSV) attribute is mapped to CSV file column header.Input value should

be taken from csv file, So, value of the downloaded filepath which includes the file should pass in csv file.

▪ **Example:**

the user wants to download a document attached to a part as shown in the image below. The Download command is located in the action toolbar menu in the UI and it's label is "Download". So he/ she will have to specify the value "Download" for the attribute commandlabel.

`<DownloadFileUsingCommand commandlabel="Download"/>`



**Note:**

If the action toolbar menu has a sub menu, then the user will have to specify the commandlabel in a pipe separated format.

**Example:**

`<DownloadFileUsingCommand commandlabel="Download"/>`

---

Created with the Personal Edition of HelpNDoc: [Easily convert your WinHelp HLP help files to CHM with HelpNDoc's step-by-step guide](#)

---

## DownloadFileUsingIcon

### 34. DownloadFileUsingIcon:

Download file using icon from list in indented table. Prior to this tag SelectIndentedTableRow tag will have to be used to determine the row to select for download.

(id/ input) is mandatory from 22xfd06 for validating the downloaded file.

**If user don't want to check Download File then no need to provide id/input , it will automatically check the download folder specifies as 3dx-tas.download.filepath key in 3DX\_TAS.properties.**

▪ **Syntax:**

`<DownloadFileUsingIcon position="" refid="" />`

▪ **Attribute:**

- **position:(mandatory)**

This attribute specifies the column number of the concerned cell of the

row specified by SelectIndentedTableRow tag in which the download file icon is in the indented table.

- **refid: (mandatory)**

This attribute refers to the id attribute in the SelectIndentedTableRow tag and must be the same as it.

- **wait: (optional)**

This attribute contains wait time in milliseconds to find element. Default value will be taken from '3dx-tas.execution.step.timeout'. Preferred wait will be based on the time it takes to download the whole file.

- **checkDownloadFile: (optional)**

This attribute is used for activating the download file validation, possible values are true, false. default value is false.

- **input: (optional)**

This attribute is used to provide a hard-coded value for input. value of the downloaded filepath which includes the file should pass as input.

- **id: (optional)**

id(CSV) attribute is mapped to csv file column header. Input value should be taken from csv file, So, value of the downloaded filepath which includes the file should pass in csv file.

- **Attribute:**

In the image below the user wants to download file using the download icon in the Actions column of the indented table. By inspecting the DOM, the position of the column is found to be 8. Row has already been selected using SelectIndentedTableRow tag. So the tag will be:

The screenshot shows a software interface with a table. The table has columns for Name, Lock, Row Number, Title, Rev, Ver, Type, Actions, and Description. In the second row, the 'Actions' column contains a download icon (a blue square with a white downward arrow). A red box highlights this icon. Below the table, a developer tools window is open, showing the HTML structure of the table row. The 'Actions' column is highlighted with a red border. The developer tools also show the CSS styles applied to the table elements, including the position of the 'Actions' column.

```

<tr class="mx_rowSelected" style="background-color: #288fd1; color: white;">
 <td style="text-align: center;">✓
 <td style="text-align: center;">Name ▲
 <td style="text-align: center;">
 <td style="text-align: center;">Actions
 <td style="text-align: center;">

<tr>
 <td style="text-align: center;">✓
 <td style="text-align: center;">DOC-0000036
 <td style="text-align: center;">0/1
 <td style="text-align: center;">
 <td style="text-align: center;">DOC-00...
 <td style="text-align: center;">0
 <td style="text-align: center;">1
 <td style="text-align: center;">Document
 <td style="text-align: center;">
 ⬇
 Download


```

*<DownloadFileUsingIcon position="8" refid="id\_in\_SelectIndentedTableRow"/>*

---

Created with the Personal Edition of HelpNDoc: [Effortlessly create a professional-quality documentation website with HelpNDoc](#)

---

## **DragAndDrop**

### **35. DragAndDrop:**

With the help of this tag, user should be able to drag and drop the elements and scroll bar from source to target position as per inputs provided by user.

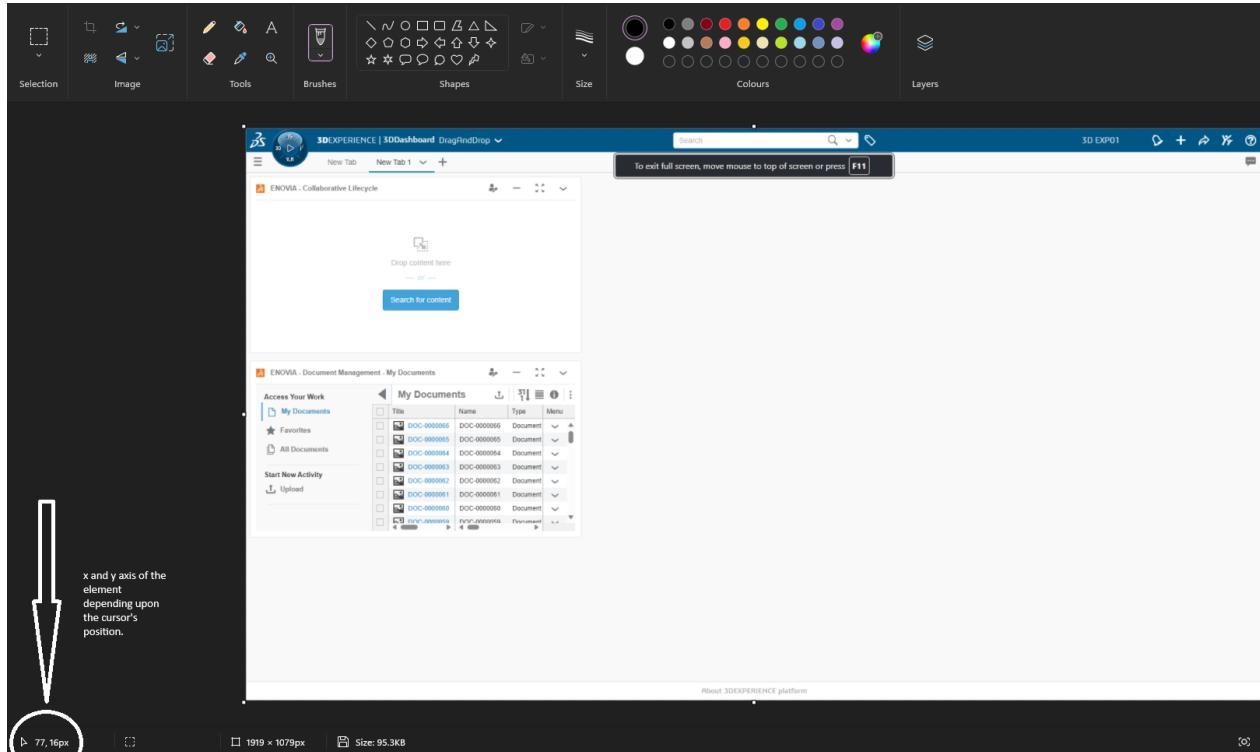
- **Syntax:**

```
<DragAndDrop sourceRefId="" sourceLocatorImage="" targetRefId=""
targetLocatorImage="" targetOffset="" />
```

#### **NOTES:**

1. Drag and drop source to target using given locatorType,sourcelocatorExpression and targetlocatorExpression attributes. For Parallel Execution DragAndDrop will not work if mode is robot, as it is a Technical Limitation .
2. while using mode ="robot" , the screen resolution should be 1920 \* 1080 and scale should be 100%.
3. While performing DragAndDrop using mode="robot" , Automation window should not be switched.
4. **For robot mode , user can use sourceOffsetX, sourceOffsetY, targetOffsetX, targetOffsetY to perform DragAndDrop without defining locatorType, sourceLocatorExpression , targetLocatorExpression.**
5. To take the value of OffSet attributes , the screen resolution should be 1920 \* 1080 and scale should be 100%, put Browser window on full screen by pressing F11 & then take a snapshot of the whole targeted window and open the snapshot in MS Paint.

Find the below MS paint image for the reference & the highlighted area is the X & Y axis of the source & targeted element.



#### ▪ **Attributes:**

- **locatorType:** (mandatory if mode is not robot)  
possible values are 'id', 'name', 'css' and 'xpath'. By using one of this Type user can give required expression. When mode is robot and sourceOffsetX, sourceOffsetY, targetOffsetX, targetOffsetY are used then locatorType becomes optional.
- **sourceLocatorExpression:** (mandatory if mode is not robot)  
defines the source from where file or object needs to be Drag. When mode is robot and sourceOffsetX, sourceOffsetY, targetOffsetX, targetOffsetY are used then sourceLocatorExpression becomes optional.
- **targetLocatorExpression:** (mandatory if mode is not robot)  
defines the target where file or object needs to be Drop. When mode is robot and sourceOffsetX, sourceOffsetY, targetOffsetX, targetOffsetY are used then targetLocatorExpression becomes optional.
- **mode:(Optional)**
  - defines the mode of the drag and drop operation. Default value is 'js'.
  - Possible values are 'selenium', 'robot', 'js' and 'enterprise'. when mode is enterprise , both sourceFrameRefId and targetFrameRefId were required.
    - 1.Mode ="js"** :::: DragAndDrop will perform only for common frame , DragAndDrop in different frames is not possible.Attributes sourceFrameRefId & targetFrameRefId will be ignored.
    - 2.Mode ="enterprise"** :::: DragAndDrop will perform only on different frames .Attributes sourceFrameRefId & targetFrameRefId is reruired.it can not be null or empty for this mode.
    - 3.Mode ="selenium"** :::: DragAndDrop will perform only for common frame , DragAndDrop in different frames is not possible.Attributes sourceFrameRefId & targetFrameRefId will be ignored.
    - 4.Mode ="robot"** :::: DragAndDrop will perform on different/common frames .Attributes sourceFrameRefId & targetFrameRefId is reruired based on the scenario.

- **sourceOffsetX:(Optional)**  
applicable only if mode is "robot". It defines the offset value for target element from top
- **sourceOffsetY:(Optional)**  
applicable only if mode is "robot". It defines the offset value for target element from top
- **targetOffsetX:(Optional)**  
applicable only if mode is "robot". It defines the offset value for target element from top
- **targetOffsetY:(Optional)**  
applicable only if mode is "robot". It defines the offset value for target element from top
- **sourceFrameRefId:(Optional)**  
applicable only if mode is 'robot' or 'enterprise'. It takes the frame refid for source element.
- **targetFrameRefId:(Optional)**  
applicable only if mode is 'robot' or 'enterprise'. It takes the frame refid for target element.

**EXAMPLE:**

```

<DragAndDrop mode="robot"
 targetLocatorExpression="//div[@class='wux-layouts-gridengine-
 poolcontainer-rel']//div[@class='wux-controls-abstract wux-layouts-collectionview-
 cell wux-cell- responsivetile'][@cell-id='3']"
 sourceLocatorExpression="//div[@class='wux-tweakers wux-tweakers-string wux-
 tweakers-string-label wux-tweakers-
 labelonly']//span[@class=' search_item_hyperlink']" locatorType="xpath"
 targetFrameRefId="id" />

<DragAndDrop locatorType="xpath"
 sourceLocatorExpression="//div[@class='wux-tweakers wux-tweakers-string
 wux-tweakers-string-label wux-tweakers-labelonly' and text()='DOC-0000066']"
 targetLocatorExpression="//div[@class='searchView']//div[@class='sentence-
 text wux-icon-gray' and text()='Drop content here']"
 sourceFrameRefId="sourceIframe1" targetFrameRefId="toIframe2"
 sourceOffsetX="483" sourceOffsetY="565" targetOffsetX="278"
 targetOffsetY="252"
 mode="robot" />

<DragAndDrop locatorType="xpath"
 sourceFrameRefId="sourceIframe1" targetFrameRefId="toIframe2"
 sourceOffsetX="483" sourceOffsetY="565" targetOffsetX="278"
 targetOffsetY="252"
 mode="robot" />

<DragAndDrop locatorType="xpath"
 sourceLocatorExpression="//div[@class='wux-tweakers wux-tweakers-string
 wux-tweakers-string-label wux-tweakers-labelonly' and text()='DOC-0000066']"
 targetLocatorExpression="//div[@class='searchView']//div[@class='sentence-
 text wux-icon-gray' and text()='Drop content here']"
 sourceFrameRefId="sourceIframe1" targetFrameRefId="toIframe2"
 mode="robot" />

<DragAndDrop mode="selenium" targetLocatorExpression="//*[@id='tasks-
 stateForDisplay-In Progress']/div[@class='body ui-droppable']"
 locatorType="xpath"
 sourceLocatorExpression="//div[contains(text(), 'abc')]/../../.."/>

```

```

<DragAndDrop mode="js" targetLocatorExpression="//*[@id='tasks-stateForDisplay-In Progress']/div[@class='body ui-droppable']"
locatorType="xpath"
sourceLocatorExpression="//div[contains(text(), 'abc')]/../../.."/>

<DragAndDrop locatorType="xpath"
sourceLocatorExpression="//div[@class='wux-tweakers wux-tweakers-string-wux-tweakers-string-label wux-tweakers-labelOnly' and text()='DOC-0000066']"
targetLocatorExpression="//div[@class='searchView']//div[@class='sentence-text wux-icon-gray' and text()='Drop content here']"
sourceFrameRefId="sourceIframe1" targetFrameRefId="toIframe2"
mode="enterprise" />

```

**NOTE:**

*Alternative for DragAndDrop tag :If drag and drop tag is not working then we can go for Action tag with mouseDown and mouseUp attribute,  
it's use with FindElement tag. Please check below Example: "**Action tag with FindElement tag**"*

- **mousedown:**

This attribute is used for **MouseDown** action with findElement tag and it's alternative for DranAndDrop tag. (Drag means mouseDown Oration)  
It's click the **mouse Right button for down action.**

- **mouseup:**

This attribute is used for **MouseUp** action with findElement tag and it's alternative for DranAndDrop tag. (Drop means mouseUp Oration)  
It's click the **mouse Right button for UP action.**

- **Examples:**

- **Iframes in Drag and Drop Objects:**

```

<FindElement locatorType="xpath" locatorExpression="//div[@class='wp-tab-panel selected']//div[@class='moduleContent']//iframe[contains(@src,'ENXWDODC_AP')]"
id="sourceframe1"/>
<SwitchToFrame refid="sourceframe1"/>
<!-- Fd06 Object -->
<FindElement locatorType="xpath" locatorExpression="//div[@class='wux-tweakers wux-tweakers-urlobject']//a[@class='wux-tweakers-string-label' and text()='DOC-89686518-0000211']" id="m1"/>
<Action name="mousedown" refid="m1"/>
<Wait time="1000" />
<FindElement locatorType="xpath" locatorExpression="//div[@class='wp-tab-panel selected']//iframe[contains(@src,'ENOLCMI_AP')]" id="tolframe2"/>
<SwitchToFrame refid="tolframe2"/>
<Wait time="10000" />
<FindElement locatorType="xpath"
locatorExpression="//div[@class='searchView']//div[@class='sentence-details align-center']/button" id="m2"/>
<Action name="mouseup" refid="m2"/>

```

- **Without Iframes in Drag and Drop Objects :**

```

<FindElement locatorType="xpath" locatorExpression="//div[@data-id='ENORERE_AP']"
id="m1"/>
<Wait time="1000" />
<Action name="mousedown" refid="m1"/>
<Wait time="2000" />

```

```
<FindElement locatorType="xpath"
locatorExpression="{//div[contains(@class,'moduleWrapper ')])[1]" id="m2"/>
<Action name="mouseup" refid="m2"/>
```

Created with the Personal Edition of HelpNDoc: Maximize Your Reach: Convert Your Word Document to an ePub or Kindle eBook

## DeleteDashboard

### DeleteDashboard:

To delete the current dashboard.

- **Syntax:**

```
<DeleteDashboard />
```

Created with the Personal Edition of HelpNDoc: Generate EPub eBooks with ease

## Deactive6wSearch

### Deactive6wSearch:

deactive6wSearch tag is use to Deactivate the activate 6W Search property.

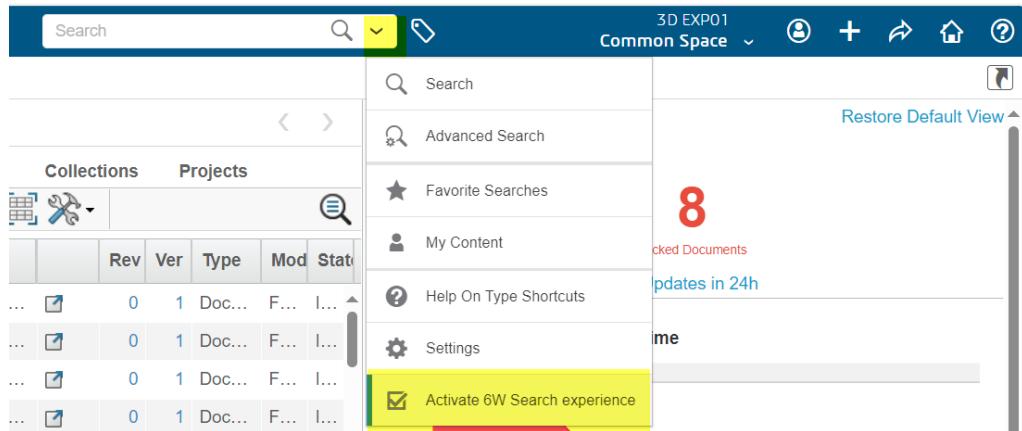
#### Attributes:

- **wait:(Optional)**

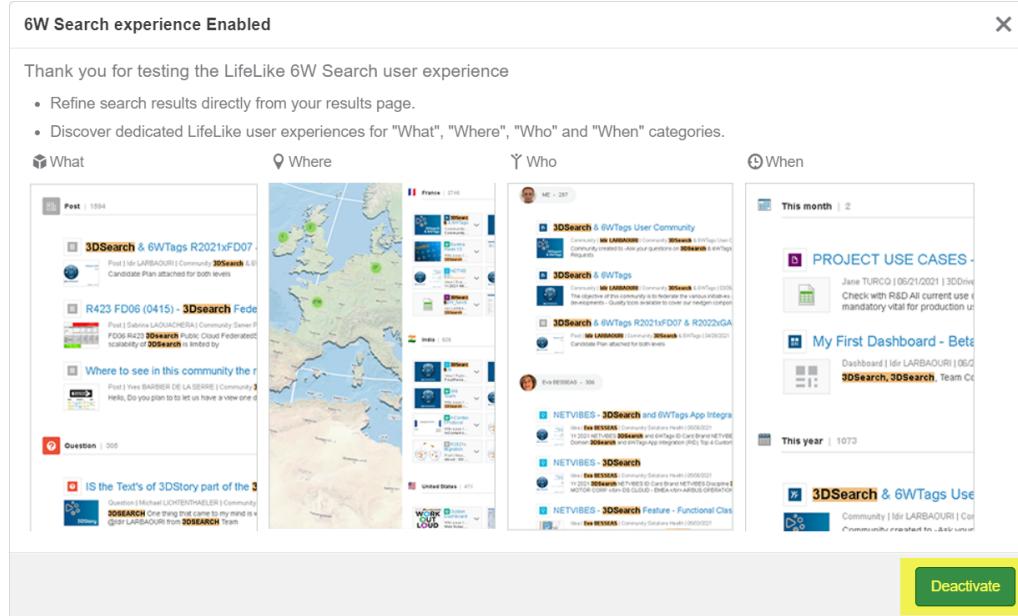
*This attribute contains wait time in milliseconds to find element. Default value will be taken from '3dx-tas.execution.step.timeout'. Preferred wait is 7000MS.*

- **Syntax:**

```
<Deactive6wSearch wait="5000"/>
```



*It will open below window and click on Highlighted button to deactivate the search experience*



*So UI will change to Normal search mode*

Created with the Personal Edition of HelpNDoc: [Simplify Your Help Documentation Process with a Help Authoring Tool](#)

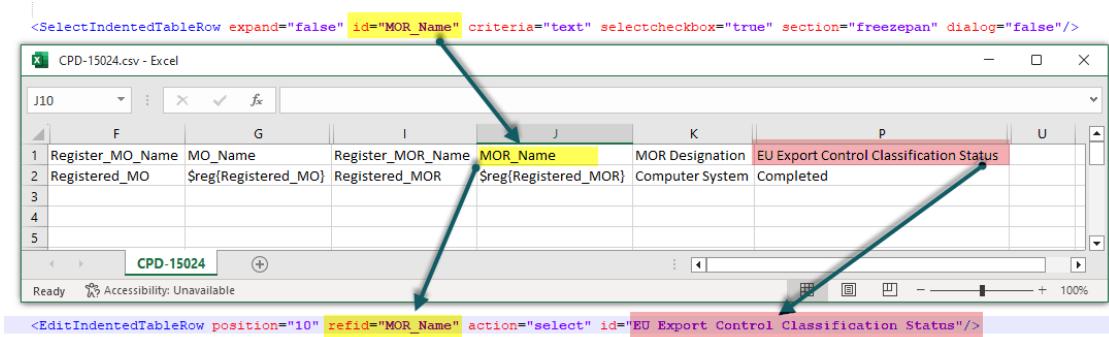
## EditIndentedTableRow

### EditIndentedTableRow:

Edit specific column of selected indented table row.

- **Syntax:** The row selection should be done using **SelectIndentedTableRow** tag.
  - <**EditIndentedTableRow** position="" refid="" action="" id="" />
    - <**SelectIndentedTableRow** expand="false" id="MOR\_Name" criteria="text" selectcheckbox="true" section="freezeapan" dialog="false"/>
    - <**EditIndentedTableRow** position="8" refid="MOR\_Name" action="text" id="ECCN Motivation"/>
    - <**EditIndentedTableRow** position="9" refid="MOR\_Name" action="select" id="EU ECCN" insideFreezePane="false" />
- **Attributes:**
  - **input:**(optional if **id** attribute is used )
    - This attribute is used to provide a hard-coded value for input instead of using id attribute from CSV file.
  - **insideFreezePane:**(optional)
    - Possible values are '**true**' or '**false**'.If criteria is true it will check inside the freeze panel.
  - **id:**(optional if **input** attribute is used)
    - id(CSV) attribute is mapped to CSV file column header.Input value should be taken from CSV file.  
So, value can be **level** or **text** should give in CSV file.
  - **refid:**(mandatory)
    - id attribute of **SelectIndentedTableRow** tag is the refid of this tag. So, using this

- refid script will identify the row to be edited.
- please check below example of using **refid** attribute it's taken the value from SelectIndentedTableRow's which is mentioned in id attribute



- **position:(mandatory)**

- It identifies the column number of the row selected using **SelectIndentedTableRow** tag to be edited.
- **Examples:** In the image below, the user wants to edit the Usage column of selected row.

It's position element can be found by inspecting the DOM and is found to be 18. So the value of position will be 18 in this case.

- **action: (mandatory)**

This attributes action to perform on specific cell of row.

Here are the possible values.

- **text:** Set the text of specific cell of given row.

- **Examples:** Please check below image. Here script will first click on qty cell of given row, then enter given text into input box.

Name	Type	Rev	Policy	F/N	Ref Des	Comp. Loc.	Description	State	Qty	U of M	Usage
A-0000104-01	Part	A	Develop...					Create		EA (each)	
A-0000118-01	Part	A	Develop...	2			Auto_Test_Part	Create	1.0	EA (each)	Standard
A-0000106-01	Part	A	Develop...	3				Create		EA (each)	

- **select:** Select item within dropdown list of specific cell of given row.
  - **Examples:** Please check below image. Here script will first click on usage cell of given row, then select specific item within dropdown list.

Name	Type	Rev	Policy	F/N	Ref Des	Comp. Loc.	Description	State	Qty	U of M	Usage
A-0000104-01	Part	A	Develop...					Create		EA (each)	
A-0000118-01	Part	A	Develop...	2			Auto_Test_Part	Create	1.0	EA (each)	Standard
A-0000106-01	Part	A	Develop...	3				Create	1.0	EA (each)	Standard

- **click:** Click on specific cell of given row.  
This action will be used when cell edit is not possible using predefined action. In other word, when we click on cell then it will show some custom edit box, in such cases user will use this action.  
This action will just click on cell and edit control of given cell will be visible. Then user can use other tags like ClickElement or SelectElement etc. to edit cell value.

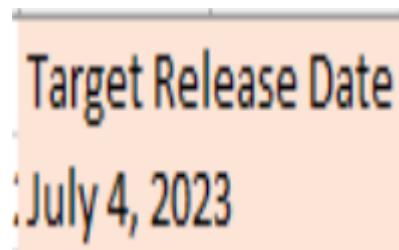
- **date:** Used to enter date in the specific cell of selected row. User must pass id attribute in CSV file for date, please check the format given below.  
In the image below, the script will click on the cell and then on the date chooser in the selected cell to enter the date.

**Examples:**

```
<SelectIndentedTableRow expand="false" id="row" criteria="level" />
<EditIndentedTableRow id="Target Release Date" action="date" />
```

refid="row" position="11/>

Name	Type	Rev	Policy	Part Family	Estimated Cost	Effectivity Date	Is Ve...
A-0000150-01	Part	A	Dev...		0.0 Dollar		FALSE
A-0000151-01	Part	A	Dev...		4.0 Pounds	Jun 19, 2018	TRUE
A-0000152-01	Part	A	Dev...		4.5 Euro	Dec 12, 2017	TRUE



- **textwithselect:** Used to enter text and select value from dropdown in the specific cell of the selected row. User has to specify the value of id attribute in a pipe separated fashion.

**Examples:**

In the image below, the Estimated Cost column will take a text value and a value to select from the dropdown.

User will have to provide these 2 values in the CSV file as

"Estimated\_Cost",  
"4.5|Euro", "0", "c"

Name	Related Physical Title	Type	Rev	Policy	Part Family	Estimated Cost	Effectivity Date	Is Version	F/N
A-0000150-01		Part	A	Dev...		0.0 Dollar		FALSE	
A-0000151-01		Part	A	Dev...		4.5	Euro		
A-0000152-01		Part	A	Dev...					

- **checkbox:** Used to select/deselect checkbox in the specific cell of the selected row. The id attribute in this case can take dummy value as it is not required to populate any data in the concerned cell.

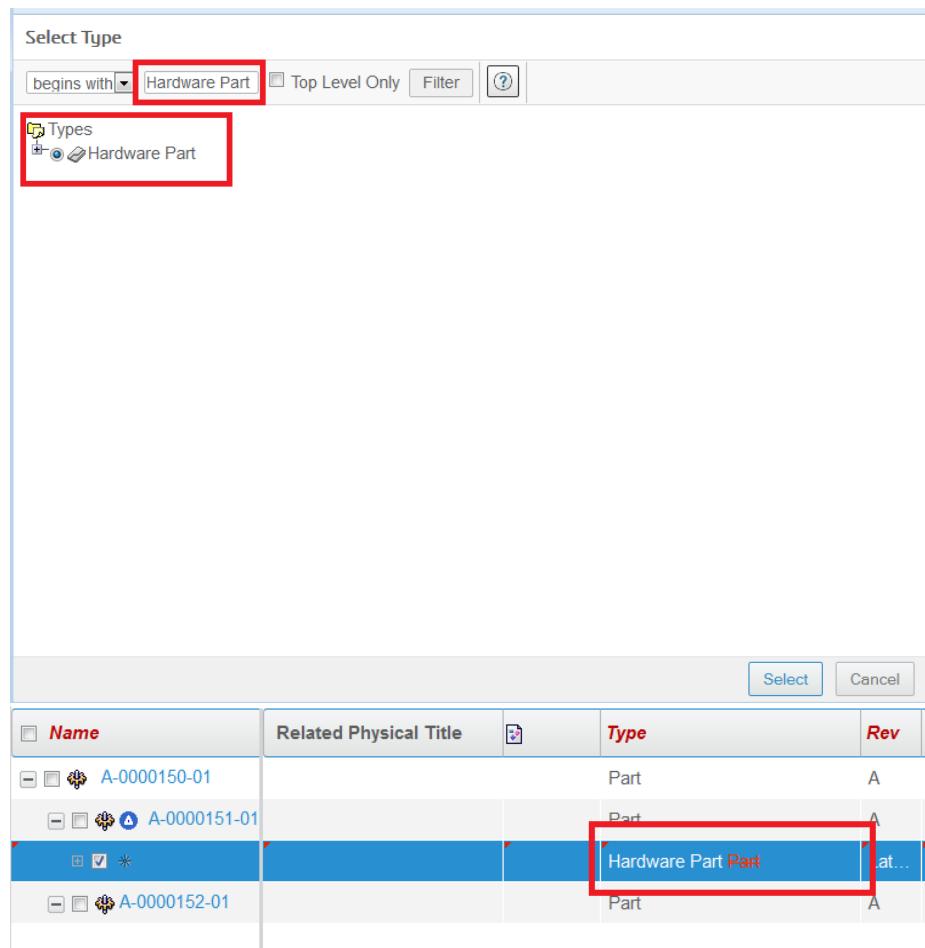
Name	Related Physical Title	Type	Rev	Policy	Part Family	Estimated Cost	Effectivity Date	Is Version	F/N
A-0000150-01		Part	A	Dev...		0.0 Dollar		FALSE	
A-0000151-01		Part	A	Dev...		4.0 Pound	Jun 19, 2018	<input checked="" type="checkbox"/>	1
A-0000152-01		Part	A	Dev...		4.5 Euro	Dec 12, 2017	TRUE	2

- **typechooser:** Used to select type to populate in the specific cell of the selected row. Id attribute takes the value to search and select from the CSV file.

**Examples:**

In the images below the script will click on the Type cell, then launch the type chooser, search for the value obtained from the CSV file and then choose it for the Part in concern.

Name	Related Physical Title	Type	Rev	Policy	Part Family
A-0000150-01		Part	A	Dev...	
A-0000151-01		Part	A	Dev...	
*		Part			Part

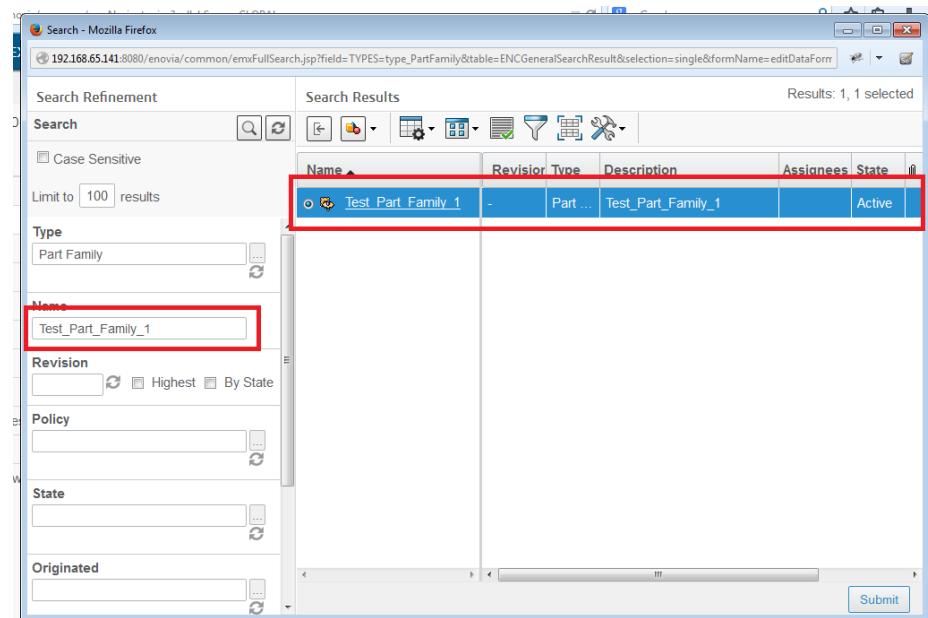


- **chooser:**

Used to search for and choose value for the specific cell of the selected row. Id attribute takes the value to search and select from the CSV file.

**Examples:**

In the images below the script will click on the Part Family cell, then launch the chooser, search for the value obtained from the CSV file and then select it for the Part in concern.



Name	Related Physical Title	Type	Rev	Policy	Part Family	Estimated Cost	Effectivity Date	Is Version	F/N
A-0000150-01		Part	A	Dev...		0.0 Dollar		FALSE	
A-0000151-01		Part	A	Dev...		4.0 Pound	Jun 19, 2018	FALSE	1
		Part	Lat...						
A-0000152-01		Part	A	Dev...		4.5 Euro	Dec 12, 2017	TRUE	2

Name	Related Physical Title	Type	Rev	Policy	Part Family	Estimated Cost	Effectivity Date	Is Version	F/N
A-0000150-01		Part	A	Dev...		0.0 Dollar		FALSE	
A-0000151-01		Part	A	Dev...		4.0 Pound	Jun 19, 2018	FALSE	1
		Part	Lat...		Test_Part_Family_1	1.0			
A-0000152-01		Part	A	Dev...		4.5 Euro	Dec 12, 2017	TRUE	2

Name	Related Physical Title	Type	Rev	Policy	Part Family	Estimated Cost	Effectivity Date	Is Version	F/N
A-0000150-01		Part	A	Dev...		0.0 Dollar		FALSE	
A-0000151-01		Part	A	Dev...		4.0 Pound	Jun 19, 2018	FALSE	1
		Part	Lat...						
A-0000152-01		Part	A	Dev...		4.5 Euro	Dec 12, 2017	TRUE	2

Created with the Personal Edition of HelpNDoc: Produce electronic books easily

## Execute

### Execute:

This tag is used to execute a specified method from an external java class or jar file from TAS context.

The user must have a java class or jar file that has been added to the TAS class path/build path for it to be located and the method from the class mentioned in the tag to be executed.

#### Syntax:

```
<Execute class="" method="" id="" />
```

#### Attribute:

- id:(optional)**

Id attribute is a csv column identifier which is already discussed in InputText tag. It can be used by user to input text if necessary.

- class:(mandatory)**

It defines java class or jar file that has been added to the TAS class path/build path for it to be located.

- method:(mandatory)**

The method from the class mentioned in the tag to be executed.

#### Example:

User wants to execute a method "mainMethod" from the class HelloExternalJava in the package com.test which is a package external to TAS.

The tag will be as follows:

```

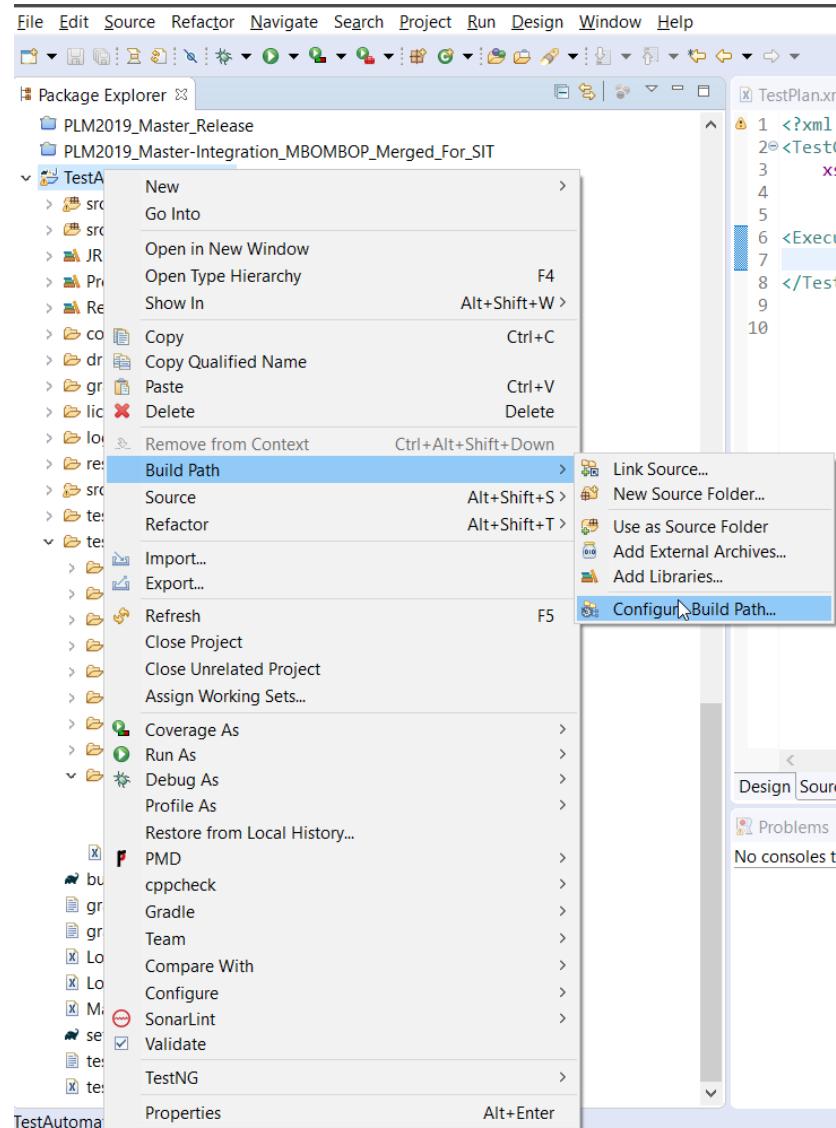
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<?testCase xmlns="https://www.steepgraph.com" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="https://www.steepgraph.com ../../resources/main/TestAutomationFramework.xsd">
 ...
 <Execute class="com.externaltest.HelloExternalJava" method="mainMethod" id="Part Pie Chart App"/>
</TestCase>

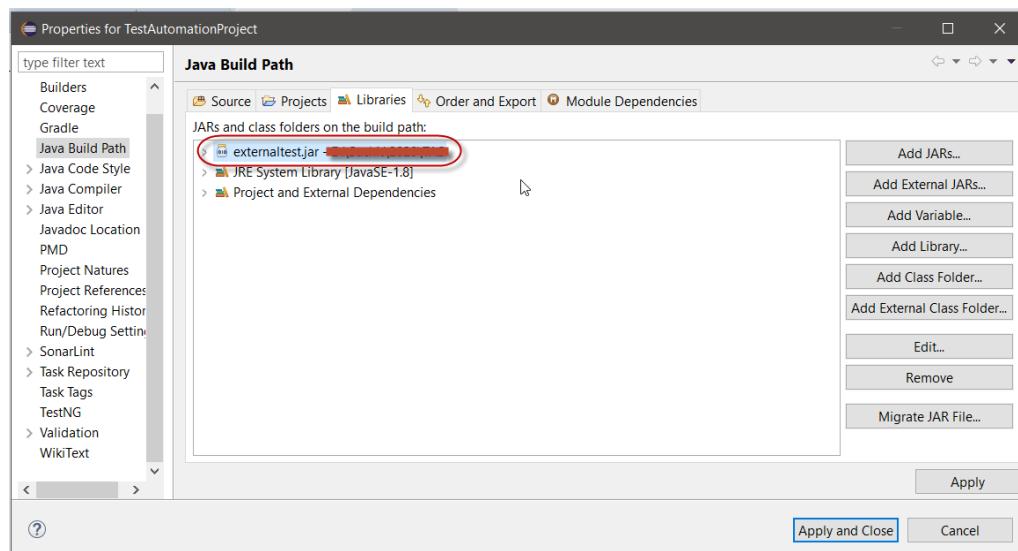
```

```
<Execute class="com.externaltest.HelloExternalJava" method="mainMethod"
id="InputText"/>
```

The java class to be executed is external to TAS and has to be present in it's build path to be executable.

From eclipse it will look like the images below. externaltest.jar file has the class and subsequent method to be executed.





When the test case is run, TAS will locate the class and execute the code in the specified method of that class.

The method code will execute accordingly.

```

1 package com.externaltest;
2
3 import java.util.Map;
4
5 import com.steepgraph.ta.framework.common.interfaces.ILibrary;
6 import com.steepgraph.ta.framework.common.pages.Driver;
7
8 public class HelloExternalJava {
9
10 public void mainMethod(Driver driver, ILibrary library, Map attrMap, String strInput) {
11 System.out.println("HELLO in com.test main....");
12 }
13
14 }

```

#### ■ Output:

```

Problems Javadoc Declaration Search Progress Error Log Git Staging Gradle Tasks Gradle Executions Console
<terminated> Launcher [Java Application] C:\Program Files\Java\jre1.8.0_152\bin\javaw.exe (26-Mar-2020, 7:54:51 pm)
3DX-TAS execution started....!
[Ljava.lang.Object;@2b4a2ec7
Started InternetExplorerDriver server (32-bit)
3.141.59.0
Listening on port 9114
Log level is set to ERROR
Only local connections are allowed
Mar 26, 2020 7:54:54 PM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: W3C
Test Case XML: testsuites\STP-21\STP-21.xml, Suite: TestPlan, Test Case: SELENIUM-440, Status: In Progress
STP-21.xml => TestPlan => SELENIUM-440 => <Execute> Processing
HELLO in com.test main...
Test Case XML: testsuites\STP-21\STP-21.xml, Suite: TestPlan, Test Case: SELENIUM-440, Status: Pass
=====
TestPlan
Total tests run: 1, Passes: 1, Failures: 0, Skips: 0
=====
Don't close the application. Please wait...
Generating Test Report...!
Test Report Generated...!
3DX-TAS Execution completed....!

```

Created with the Personal Edition of HelpNDoc: [Upgrade Your Documentation Process with a Help Authoring Tool](#)

## ELSEIF

### ELSEIF tag:

With the help of this tag, user should be able to check condition(i.e. based on text comparison and visibility). One Script may have multiple ELSEIF statement.

This is similar to ELSEIF statement in programming language. Before using this tag user need to use IF tag & after that user can use ELSEIF TAG & user can use multiple ELSEIF tag but after using the ELSEIF tag USER can use ELSE or ENDIF tag to close the whole block.

- **Syntax:**

```
<IF id="" criteria="" condition="" locatorImage="" refId="" wait="" />
```

**NOTES:**

1. The images captured for locatorImage attribute should always be in .png extension. The image captured for the required element should be focused and captured properly.
2. In ifCondition tag, the mandatory attributes are id, criteria, condition, locatorImage and refId.

- **Attributes:**

- ***id(optional)*:**

*This attribute refers to the text value of the element, which user need to compare, id(CSV) attribute is mapped to csv file column header. Input value should be taken from csv file.*

*This attribute is mandatory only when criteria is attribute or text.*

- ***locatorType(optional)*:**

*There are 4 types of locator: id, cssselector, name and xpath which defines locator type.*

- ***locatorExpression(optional)*:**

*This attribute contains expression specific to locator type.*

- ***criteria(Mandatory)*:**

*Specifies the criteria on which the element will be tested criteria can be:*

- *Text : Web element text value is equals to value passing in CSV file.*

- *attribute : check the value of attribute for Web Element Ex : class, title*

- *visible : To Check Whether or not the Web element is displayed*

- *found : Web element is found or not.*

- ***condition(optional)*:**

*There are 2 types of condition: =,!=. This attribute is only used when criteria is attribute or text. Default value id '='.*

- ***wait(optional)*:**

*This attribute contains wait time in milliseconds to find element. By default takes time from 3dx.tas.step.interval.*

- ***attribute(optional)*:**

*It is only applicable when criteria = attribute. It specifies the attribute available on web elements. E.g.- name, value, class, rmbldm, rmbrow etc. This attribute is only used when criteria is attribute.*

- ***refid:(optional):***

*id attribute of fiendElement tag will used as refid for this tag.If refId used then LocatorType & LocatorExpression will not work.*

- ***wait:(optional):***

*This attribute contains wait time in milliseconds to find element.Default value will be taken from '3dx-tas.execution.step.timeout'.Preferred wait is 6000MS.*

- ***Examples:***

Suppose user wants to compare the username text, the syntax for the same is provided below:

```
<IF locatorType="id" locatorExpression="compass_ctn" criteria="visible" id="Model" />
 <ClickElement locatorType="id" locatorExpression="compass_ctn"/>
<ENDIF/>
```

*In the above scenario, user wants to click an element if certain text is visible on a certain element. if not then will not click.*

***In below you can find the use of whole if, else, elseif, endif block .***

i) 

```
<IF id="inputtext" criteria="visible" condition="=" locatorImage="user.png"/>
 <ClickElement locatorImage="user.png"/>
<ELSEIF id="inputtext" criteria="visible" condition="!=" locatorImage="user.png"/>
 <ClickElement locatorImage="welcome.png"/>
<ELSEIF id="inputtext" criteria="visible" condition="!=" locatorImage="user.png"/>
 <ClickElement locatorImage="welcome.png"/>
<ELSE/>
 <ClickElement locatorImage="ds.png"/>
<ENDIF />
```

ii) 

```
<IF id="inputtext" criteria="visible" condition="=" locatorImage="user.png"/>
 <ClickElement locatorImage="user.png"/>
<ELSEIF id="inputtext" criteria="visible" condition="!=" locatorImage="user.png"/>
 <ClickElement locatorImage="welcome.png"/>
<ENDIF />
```

---

Created with the Personal Edition of HelpNDoc: Transform Your Word Doc into a Professional-Quality eBook with HelpNDoc

## ELSE

### Else TAG:

User can use Else when he IF & ALL the ELSEIF blocks are false.This is similar to ELSEIF statement in programming language.

Before using this tag user need to use IF or IF>ELSEIF tag & after using ELSE tag user need to use ENDIF tag.

It does not have any attribute.

***In below you can find the use of whole if, else, elseif, endif block .***

i) 

```
<IF id="inputtext" criteria="visible" condition="=" locatorImage="user.png"/>
 <ClickElement locatorImage="user.png"/>
<ELSEIF id="inputtext" criteria="visible" condition="!=" locatorImage="user.png"/>
 <ClickElement locatorImage="welcome.png"/>
```

```

<ELSEIF id="inputtext" criteria="visible" condition!="!" locatorImage="user.png"/>
 <ClickElement locatorImage="welcome.png"/>
<ELSE/>
 <ClickElement locatorImage="ds.png"/>
<ENDIF />

ii) <IF id="inputtext" criteria="visible" condition==" locatorImage="user.png"/>
 <ClickElement locatorImage="user.png"/>
<ELSEIF id="inputtext" criteria="visible" condition!="!" locatorImage="user.png"/>
 <ClickElement locatorImage="welcome.png"/>
<ENDIF />

```

Created with the Personal Edition of HelpNDoc: Streamline your documentation process with HelpNDoc's WinHelp HLP to CHM conversion feature

---

## ENDIF

### ENDIF TAG:

This tag indicate the end of IF tag. This tag will used to define bound of if statement.

*In below you can find the use of whole if, else, elseif, endif block .*

```

i) <IF id="inputtext" criteria="visible" condition==" locatorImage="user.png"/>
 <ClickElement locatorImage="user.png"/>
<ELSEIF id="inputtext" criteria="visible" condition!="!" locatorImage="user.png"/>
 <ClickElement locatorImage="welcome.png"/>
<ELSEIF id="inputtext" criteria="visible" condition!="!" locatorImage="user.png"/>
 <ClickElement locatorImage="welcome.png"/>
<ELSE/>
 <ClickElement locatorImage="ds.png"/>
<ENDIF />

ii) <IF id="inputtext" criteria="visible" condition==" locatorImage="user.png"/>
 <ClickElement locatorImage="user.png"/>
<ELSEIF id="inputtext" criteria="visible" condition!="!" locatorImage="user.png"/>
 <ClickElement locatorImage="welcome.png"/>
<ENDIF />

```

Created with the Personal Edition of HelpNDoc: Eliminate the Struggles of Documentation with a Help Authoring Tool

---

## FindElement

### FindElement:

Find the web element using given locatorType and LocatorExpression. When multiple actions have to be performed on the same element, then it's better to find the element once and perform action subsequently. This tag is only used to locate element in web, other tag will use reference of this tag and perform given action. and also used for SwitchToFrame

- **Syntax:**

*<FindElement locatorType="" locatorExpression="" id="" />*

- **Attributes:**

- **id: (Optional)**

Here id attribute is just a unique key that will referred by other tags to perform action on the web element located by this tag. while using another element then we have to use id attribute which is Mandatory for using another action like rightClick, doubleClick, etc....

- **input: (Optional)**

This attribute is used to provide a hard-coded value for input instead of using id attribute from csv file.

- **mode: (Optional)**

attribute is used to select execution action, viz. selenium/js based. Takes values selenium or js.

- **locatorType:(Mandatory)**

There are 4 types of locator: id, cssselector, name and xpath which defines locator type.

- **locatorExpression:(Mandatory)**

locatorExpression attribute is mandatory. This attribute contains expression specific to locator type

- **Examples:**

```
<!-- This 1st Scenario -->
<FindElement locatorType="xpath" locatorExpression="//span[text() = 'Collaboration and Approvals']" id="actionRightClick"/>
<Action name="rightclick" refid="actionRightClick"/>

<!-- This 2nd Scenario : Change frame APPID: from CSV file-->
<FindElement locatorType="xpath"
locatorExpression="//iframe[contains(@src, '$csv{APPID}')]" id="Frame"/>
<SwitchToFrame refid="Frame"/>
```

For example , If user want to find "OK" button and want to perform any action like click then this can be used.

```
<FindElement locatorType="xpath" locatorExpression="//button[@class='btn-primary']" id="Ok"/>
```

Create Change Order |

Action  

Description  
Act

Organization  
Company

Reported Against  [Clear](#)

Change Coordinator  [Clear](#)

Category of Change  
Unassigned

Severity  
Low

Due Date  [Clear](#)

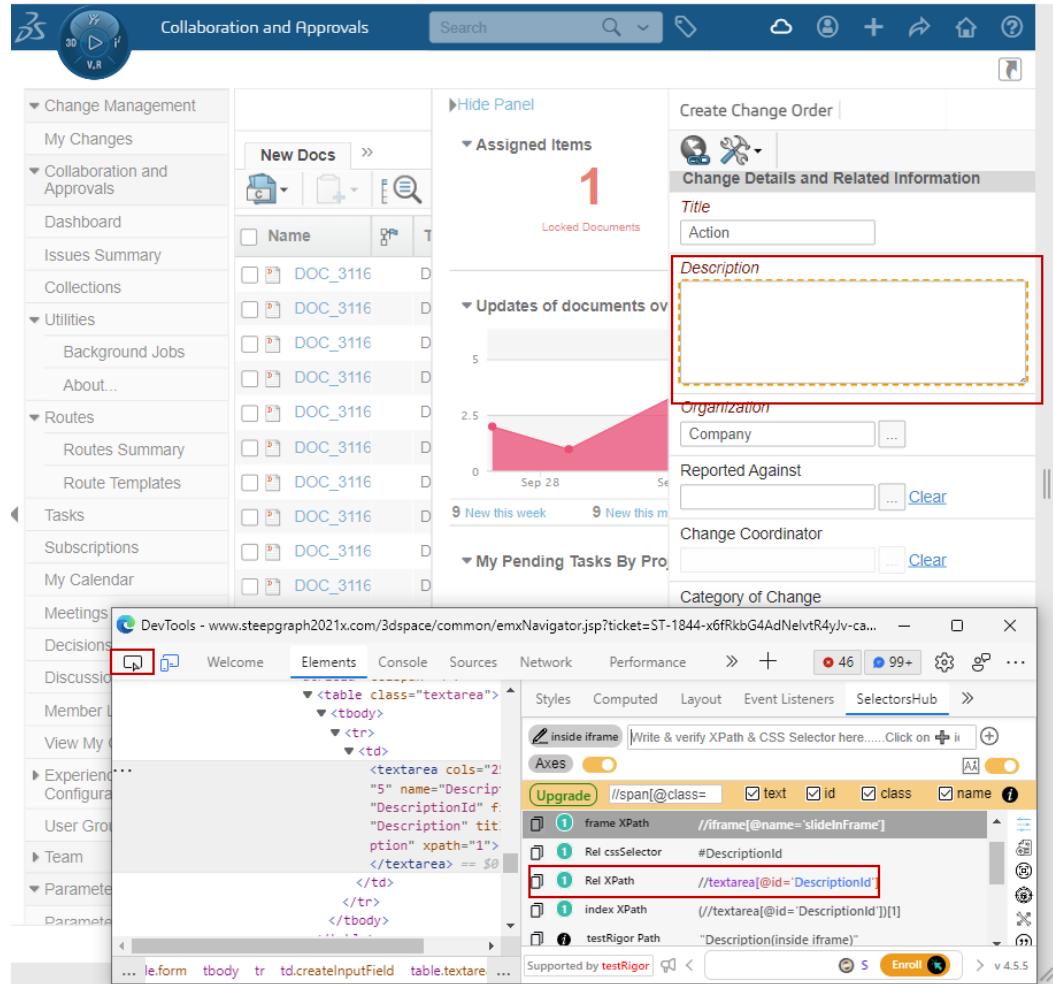
**Reviewers and Approvers**

Reviewer List  [Clear](#)

Approvers  [Clear](#)

**Extended Attributes**

Informed Users  [Clear](#)



**2.** If user wants to find element and wants to give input then input attribute along with find element tag.

*<FindElement*

*locatorType="xpath" locatorExpression="//textarea[@id='DescriptionId']" input="Action"/>*

To find element user should give x path for that element by inspecting DOM and input should be given in the hard coded value. Here E.g for Description Box is given .

---

Created with the Personal Edition of HelpNDoc: [Effortlessly Create Professional Documentation with HelpNDoc's Clean UI](#)

---

## FilterSearchForm

### FilterSearchForm:

User can use **FilterSearchForm** tag when use choosing advance search type then this tag will used for filling the Search filter form.

#### ▪ Syntax:

```
<FilterSearchForm inputfieldtype="" fieldlabel="" id="" selection="" criteria ="" />
```

#### ▪ Attribute:

- **inputFieldType: (mandatory)**

There are 3 types of values:

- **text** : User can use this text value after field label attribute.
- **date** : User can use this date value after field label attribute with

FromDate orToDate or between value. provide date from **CSV** with id attribute

- **select** : User can use this text value after field label attribute with full value or pipe separated values.
- **textChooser**: This attribute is now Deprecated, it was used in 17x only.
- **Object**: This attribute is now Deprecated, it was used in 17x only.
- **person**: This attribute is now Deprecated, it was used in 17x only.
- **click**: This attribute is now Deprecated, it was used in 17x only.

- **fieldLabel:(mandatory)**

This attribute used for filed label and specified the label on UI.

- **id:(mandatory)**

(CSV) attribute is mapped to CSV file column header, id attribute is used to take value to search from CSV.

- **selection (optional)**

This attribute is used to select the value from searched value on UI, just you have specified to give choices value like single, multiple checkbox.

single means : it will select only single checkbox.

multiple means : it will select only multi checkbox-es.

- **Criteria:(optional)**

This attribute is used to select date criteria, **NOTE**: Date format is 28-11-2023 use this format otherwise tag will failed.

1. **on** : From & To : Same Date only.

2. **on or after** : From Date only .

3. **on or before** : To Date only.

4. **between** : From and ToDates. It will take in Pipe separator in id(CSV): "12-12-23|16-11-23".

- **row : (mandatory)** : This attribute used to provide the single and multiple value in pipe separator in row selection .

- **Example:**

```

<FilterSearchForm inputfieldtype="date" fieldlabel="Modification
Date" id="DateValue" criteria="on"/>

<!-- <FilterSearchForm inputfieldtype="date"
fieldlabel="Modification Date" id="DateValue" criteria="on or after"/> -->

<!-- <FilterSearchForm inputfieldtype="date"
fieldlabel="Modification Date" id="DateValue" criteria="on or before"
selection="single"/> -->

<!-- <FilterSearchForm inputfieldtype="date"
fieldlabel="Modification Date" id="DateValue" criteria="between"/> -->

<!-- <FilterSearchForm inputfieldtype="select" fieldlabel="Type"
id="TypeValue" selection="multiple" /> -->

<!-- <FilterSearchForm inputfieldtype="text" fieldlabel="Type"
id="TypeValue" /> -->

```

The screenshot shows the '3DEXPERIENCE | ENOVIA Collaboration and Approvals' interface. On the left, there is a sidebar with fields for 'Source' (Collaborative Spaces), 'Type' (Select a type (optional)), 'Extension' (No subtype for selected type), 'Title' (Enter text), 'Name' (Enter text), and 'Modification Date' (From and To). On the right, there is a search results table titled '3DSearch - Advanced Searches' with columns for 'Title', 'Type', 'Description', and 'Name'. A search bar at the top right says 'Enter a keyword to perform a search'. Below the table, there is a message '216 Results'.

This screenshot shows the same interface as above, but the search results table is more detailed. It includes additional columns: 'Modification Date' and 'Creation Date'. The table lists 216 results, each row containing a small icon, the title, type, name, modification date, creation date, and revision information. The 'Selected Tags' sidebar on the left now includes 'Physical Product'.

Created with the Personal Edition of HelpNDoc: [Eliminate the Struggles of Documentation with a Help Authoring Tool](#)

## FilterSearchPerson

### FilterSearchPerson:

User can use **FilterSearchPerson** tag when use **ellipse(...)** section and provide the label name as per the action. It's Search the value from CSV value, then it will select the text and row number.

- **Syntax:**

```
<FilterSearchPerson row="" fieldlabel="" id="" selection="" />
```

- **Attribute:**

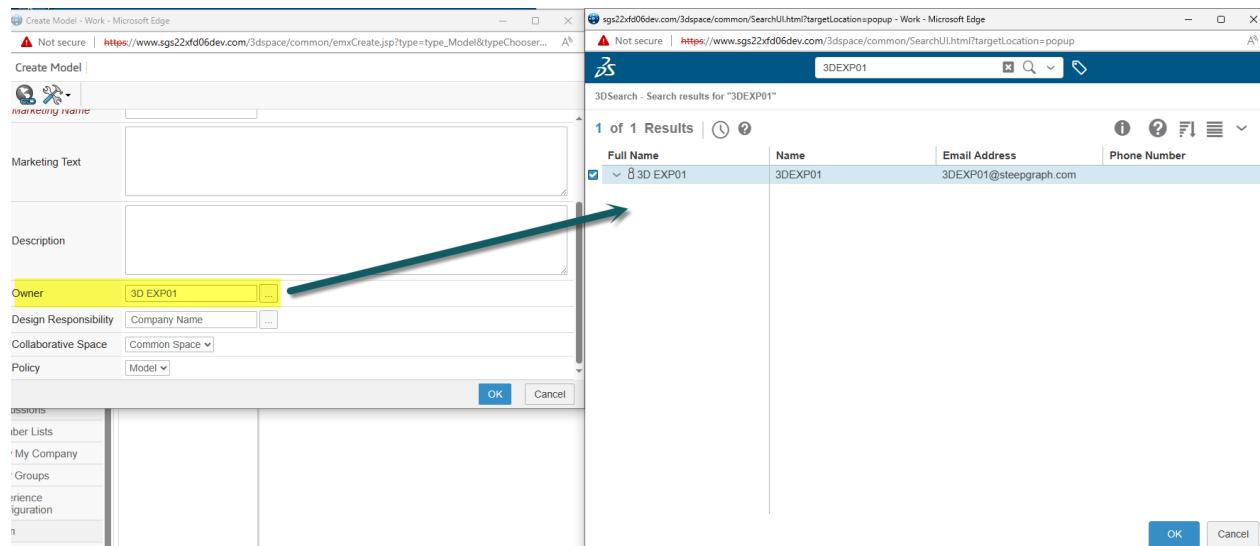
- **fieldLabel:(mandatory)**

This attribute used for filed label.

- **id:(mandatory)**  
(mandatory) :id(CSV) attribute is mapped to csv file column header, id attribute is used to take value to search from csv
- **selectionType:(mandatory)**  
There are 2 types of values:
  - **text** : User can use this text value after field label attribute.
  - **row-number** : User can use this attribute when text is not working, you have to row attribute for providing row number.
- **row: (optional)**  
This attribute is used for row number.

▪ **Example:**

```
<Login username="$csv{username}" password="$csv{password}" />
<SwitchToDefaultContent/>
<Wait time="5000"/>
<ClickGlobalActionsMenu commandLabel="Product Line/Create Model"/>
<SwitchToWindow/>
<FilterSearchPerson row="1" selection="row-number" fieldlabel="Owner"
id="username" />
```




---

Created with the Personal Edition of HelpNDoc: [Easily create CHM Help documents](#)

## GlobalSearch

### GlobalSearch:

*With the help of this tag, user should be able to search object in 3DX Native Application.*

▪ **Syntax:**

```
<GlobalSearch id="" type="" searchtype="" />
```

#### NOTES:

1. The id attribute is mandatory and rest of the attributes(i.e. type and searchtype are optional).
2. use either **id** or **input** attribute for this tag.

▪ **Attributes:**

- ***id:(Optional when input used)***

*This attribute refers to the text, which needs to be searched by the user and the same should be provided in csv. This attribute id is mapped to csv column header. User should provide the column header in the script, wherever, he needs to provide the reference of this.*

- ***input :(Optional when id used)***

*attribute is used to provide a hard-coded value for input instead of id attribute from csv file.*

- ***highlight : (optional)***

*attribute used is choose whether to highlight the element when working on it.*

*Takes values true or false.*

- ***type:(optional)***

*This attribute refers to type from global search type. The default type value for this attribute is All.*

- ***style:(optional)***

*attribute is used to provide styling to the highlight used.*

- ***searchtype:(optional)***

*This attribute refers to search type, whether user wants to perform normal search or Advanced search. The possible values for this attribute is either Search or Advance Search. The default searchtype is Search.*

▪ **Example:**

Support user like to search A-0000104-01 Part in 3DEXperience. First, he/she need to define column in csv and enter its value as "A-0000104-01".

```
"PartName", "Qty"
"A-0000104-01", "1"
```

Then define tag in test case like

```
<GlobalSearch id="PartName" type="All" searchtype="Search"/>
```

Then you will see the global search result like

Search Refinement

Search A-0000104-01

Case Sensitive Limit to 100 results

Type

Revision Highest By State

Policy

State

Originated

Title

Modified

Search Results

Display Name	Revisor	Type	Description	State	Modified
A-0000104-01	A	Part		Create	28-May-2018

Created with the Personal Edition of HelpNDoc: [Keep Your Sensitive PDFs Safe with These Easy Security Measures](#)

## GetValueFromCookies

### GetValueFromCookies:

This tag is used for getting value from cookies

- **Syntax:**

`<GetValueFromCookies cookieName="" />`

where *cookieName* is the key name in cookies.

- **Attributes:**

- **cookieName:(mandatory)**

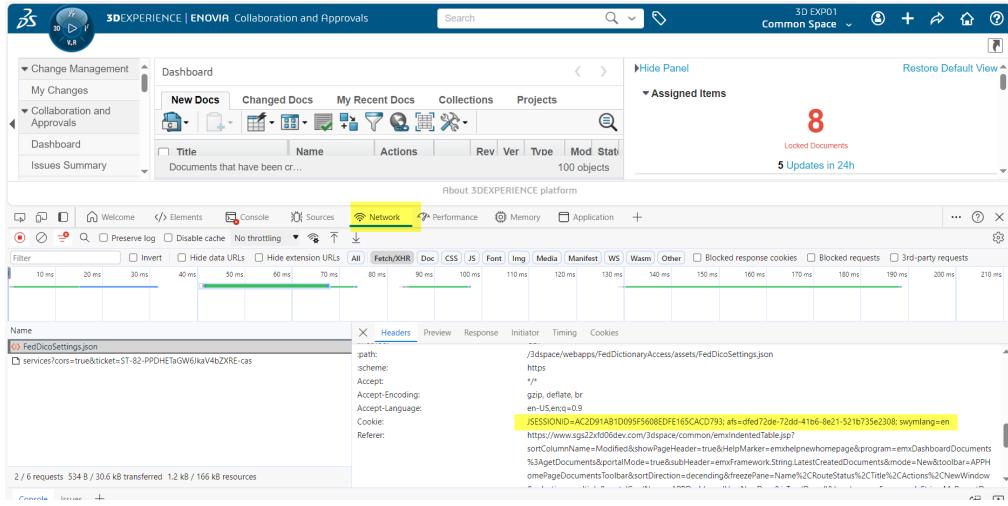
used to give name of cookie key

**Note :** cookieName should be passed as it is given in cookie key(case sensitive).

For example,

`<GetValueFromCookies cookieName="JSESSIONID" />`

*YOU will get JSESSIONID*



Created with the Personal Edition of HelpNDoc: [Easily create Qt Help files](#)

## HandleAlert

### HandleAlert:

User use this tag to handle an alert pop up message.

- **Syntax:**

```
<HandleAlert action="" />
```

- **Attributes:**

- **action(Mandatory):**

action attribute has two possible values:

1. accept - It used to accept the alert
2. dismiss- It used to denied the alert.

- **optional(optional):**

possible values are true or false.This attribute is used to handle "YES" or "NO" action.

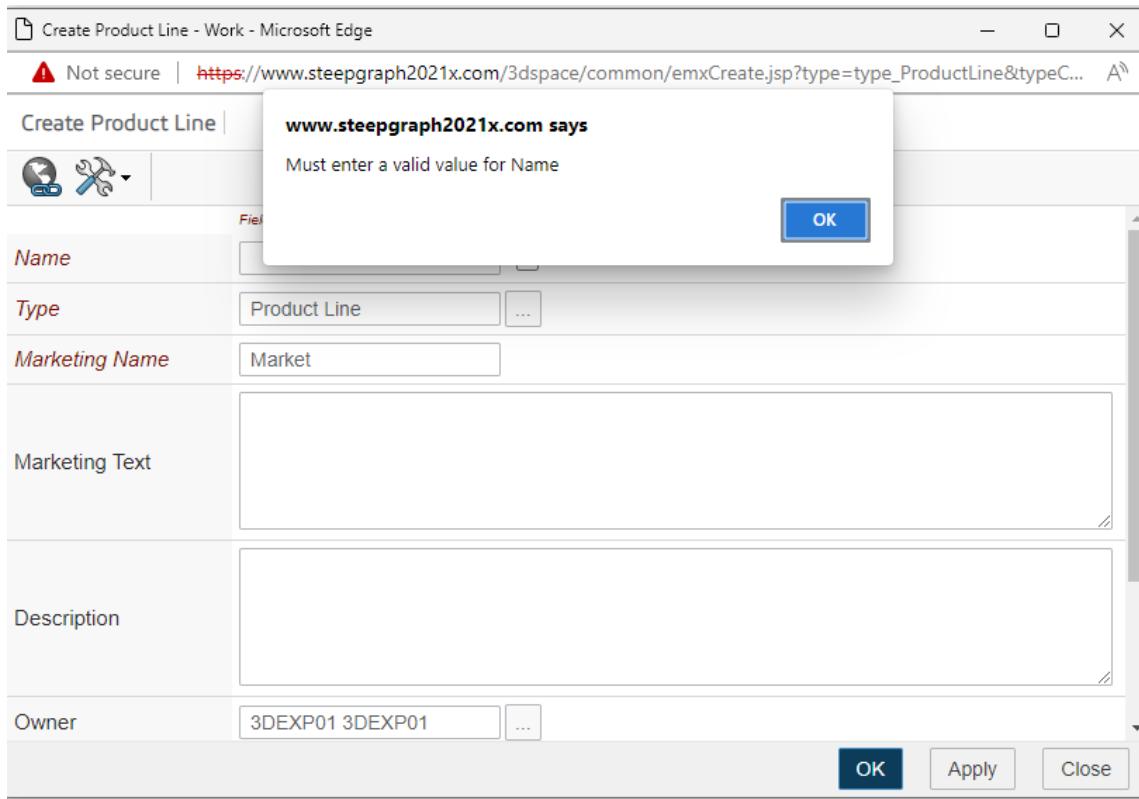
- **Examples:**

User use wants to accept or dismiss alert pop up message on screen this tag is used.

If mandatory field is not filled then after clicking on Ok button it will show alert on

screen to handle that alert this tag can be used.

```
<HandleAlert action="accept" />
```



Created with the Personal Edition of HelpNDoc: [Elevate Your Help Documentation with a Help Authoring Tool](#)

## HighlightElement

### HighlightElement :

Highlight Element is used to highlight the located expression with some style.

- **Syntax:**  
`<HighlightElement locatorType="id" locatorExpression="" id="" />`
- **Attributes:**
  - **locatorExpression:(mandatory)**  
 This attribute contains expression specific to locator type
  - **locatorType:(mandatory)**  
 There are 4 types of locator: id, cssselector, name, and xpath which defines locator type
  - **id:(optional)**  
 Here id attribute is just a unique key that will be referred by other tags to perform action on the web element located by this tag.
  - **style:(optional)**  
 This attribute is used to provide styling to the highlight used. Default value is taken from 3DXTAS Property key '[3dx-tas.highlight.webelement.style](#)'.
- **Examples:**  
 In the image below, if the user wants to Highlight Element by using located expression then the tag will be:

*<HighlightElement locatorType="id" locatorExpression="DescriptionId" id="DescriptionId"/>*

Created with the Personal Edition of HelpNDoc: [Maximize Your Productivity with HelpNDoc's Efficient User Interface](#)

## IF

### IF tag:

With the help of this tag, user should be able to check condition(i.e. based on text comparison and visibility). One Script may/may not have multiple IF statement.

This is similar to IF statement in programming language. If the user only use IF tag when after the IF block User need to use ENDIF tag to completely close the clock of IF TAG.

- **Syntax:**

*<IF id="" criteria="" condition="" locatorImage="" refId="" wait="" />*

**NOTES:**

1. The images captured for locatorImage attribute should always be in .png extension. The image captured for the required element should be focused and captured properly.
2. In ifCondition tag, the mandatory attributes are id, criteria, condition, locatorImage and refId.

- **Attributes:**

- ***id(optional)*:**

*This attribute refers to the text value of the element, which user need to compare, id(CSV) attribute is mapped to csv file column header. Input value should be taken from csv file.*

*This attribute is mandatory only when criteria is attribute or text.*

- ***locatorType(optional)*:**

*There are 4 types of locator: id, cssselector, name and xpath which defines locator type.*

- ***locatorExpression(optional):***  
*This attribute contains expression specific to locator type.*
  - ***criteria(Mandatory):***  
*Specifies the criteria on which the element will be tested*  
*criteria can be:*
    - *Text : Web element text value is equals to value passing in CSV file.*
    - *attribute : check the value of attribute for Web Element Ex : class, title*
    - *visible : To Check Whether or not the Web element is displayed*
    - *found : Web element is found or not.*
  - ***condition(optional):***  
*There are 2 types of condition: =,!=.This attribute is only used when criteria is attribute or text.Default value id '='.*
  - ***wait(optional):***  
*This attribute contains wait time in milliseconds to find element.Default value will be taken from '3dx-tas.execution.step.timeout'.Preferred wait is 6000MS.*
  - ***attribute(optional):***  
*It is only applicable when criteria = attribute.It specifies the attribute available on web elements.E.g.- name, value, class, rmbldm, rmbrw etc.This attribute is only used when criteria is attribute.*
  - ***refid:(optional):***  
*id attribute of friendElement tag will used as refid for this tag.If refid used then LocatorType & LocatorExpression will not work.*
  - ***wait:(optional):***  
*This attribute contains wait time in milliseconds to find element.Default value will be taken from '3dx-tas.execution.step.timeout'.Preferred wait is 6000MS.*
- ***Examples:***  
Suppose user wants to compare the username text, the syntax for the same is provided below:
- ```
<IF locatorType="id" locatorExpression="compass_ctn" criteria="visible" id="Model" />
    <ClickElement locatorType="id" locatorExpression="compass_ctn"/>
<ENDIF/>
```
- In the above scenario, user wants to click an element if certain text is visible on a certain element. if not then will not click.*
- In below you can find the use of whole if, else, elseif, endif block .***
- i) <IF id="inputtext" criteria="visible" condition="=" locatorImage="user.png"/>
<ClickElement locatorImage="user.png"/>
<ELSEIF id="inputtext" criteria="visible" condition="!=" locatorImage="user.png"/>
<ClickElement locatorImage="welcome.png"/>
<ELSE/>

```
<ClickElement locatorImage="ds.png"/>
<ENDIF />
```

Created with the Personal Edition of HelpNDoc: [Full-featured EBook editor](#)

Include

Include:

With the help of this tag, user could include the created building blocks to scripts.

- ***Building Block:***

1. Each building block consists of simple xml script.
2. All the building blocks will be placed in an test suites folder, We can create the separate folder for building blocks E.g: BB
3. These will be later used to create final test script
4. We can use multiple building blocks in single test script

NOTE:

The images for the Building Block should be stored under Images folder, which is present under Building Block folder. In case user is not able to find the same, user should create one and store images in same.

- ***Syntax:***

```
<Include filepath="testsuites\BB\Login.xml"/>
```

- ***Attributes:***

- **filepath(Mandatory):**

Path of the building block (xml file) present in testsuites.

Created with the Personal Edition of HelpNDoc: [Make Help Documentation a Breeze with a Help Authoring Tool](#)

InputText

InputText

Insert a text into input field (web element) identified using given [locatorType](#) and [LocatorExpression](#) or [id of findElement used refid for this tag](#). Input value is taken from csv. id attribute is map to csv column header. Test data is taken from csv file so when user need to pass input to web element, then he/she need to define column name with same name as id of the current xml tag and pass value in respective rows of csv.

- ***Attributes:***

- **locatorExpression:(Optional if RefId used)**

This attribute contains expression specific to locator type.

LocatorExpression is mandatory attribute when **refid** is not define.

- **locatorType:(Optional if RefId used)**

There are 4 types of locator: id, cssselector, name, and xpath which defines locator type. **Locatortype is mandatory** attribute when **refid** is not define.

- **refid:(Optional of if locatorExpression & locatorType used)**

id of FindElement tag will used as refid for this tag. If locatorType and locatorExpression is not define then refid is mandatory attribute. As

shown in the below example.

```
<FindElement locatorType="xpath"
  locatorExpression="//input[@id='Name'] id="Name"/>
<InputText refid="Name" input="Market1"/>
```

- **id:(Optional)**

id attribute is mapped to csv file column header. Input value should be taken from csv file. You can pass Id value hard coded in that also

As Shown in below

- **Example:**

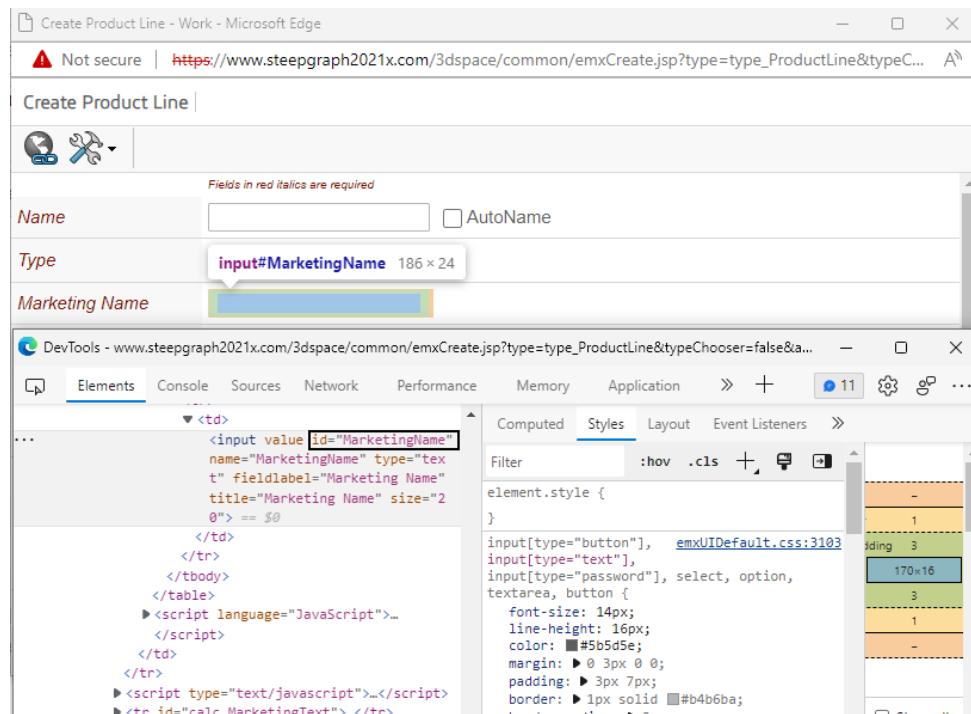
```
<InputText locatorType="id" locatorExpression="MarketingName"
```

```
input="Market1"/>
```

below You can pass this value in csv file also as shown below.

MarketingName

Market1



- **encrypted:(optional)**

This attribute is used to allow encrypted/un-encrypted text entry into input field.

Encrypted/ DecryptED value in CSV file then only it work.

There are possible two value "true" and "false" default value is false. As Shown in Below

- **Example**

```
<InputText locatorType="xpath"
```

```
locatorExpression="//textarea[@id='MarketingTextId'] id="password" encrypted="true" />
```

we need to provide this value to be encrypt in CSV file as shown below

password

T0g3aShU6v47E1iaY3JEaw==

- **highlight:(optional)**

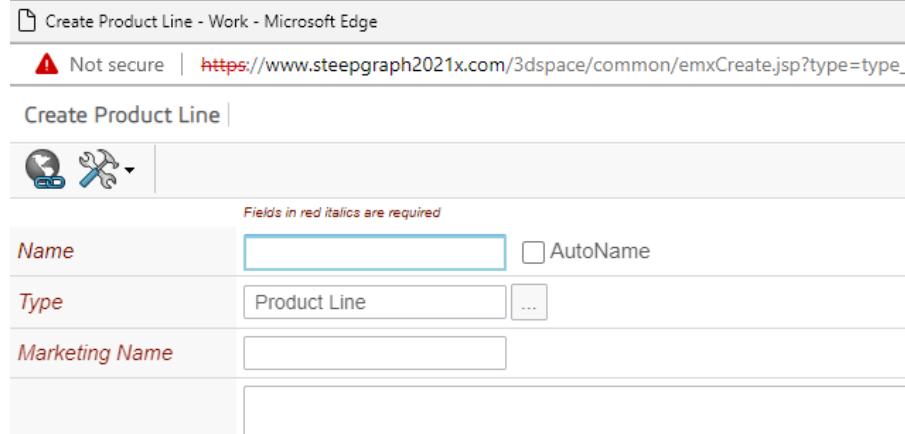
This attribute used is choose whether to highlight the element when working on it. Takes values true or false. Default value is depends on 3DX TAS

PROPERTIES KEY - **3dx-tas.highlight.webelement.enable** OR fasle if key not defined.

This Attribute will highlight the element for which you have given XPath, CSSSelector,id or name.As shown in Below

- **Example:**

```
<InputText locatorType="cssselector" locatorExpression="#Name" input="Product1" highlight="true"/>
```

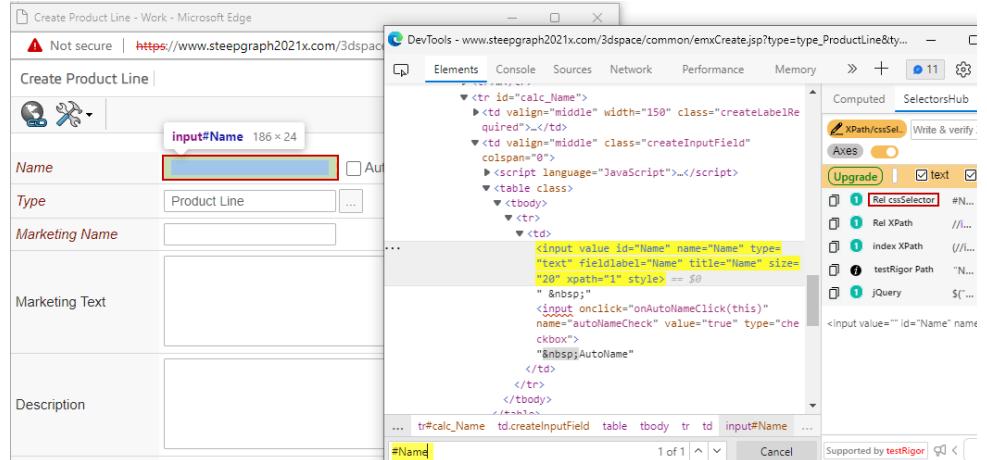


- **input:(optional)**

This attribute is used to provide a hard-coded value for input instead of using id attribute from csv file.

- **Example:**

```
<InputText locatorType="cssselector" locatorExpression="#Name" input="Product1"/>
```



2. user can use this input that when they have to write something in that field by giving it as XPath, CSSSelector, Id ,name of the field in which they want to write. As shown in the above image.

- **mode:(optional)**

This attribute is used to select execution action, viz. selenium/js based. Takes values selenium or js.if mode="js" then after the operation it will perform arrow right sendkeys operation.

- **newLine : (optional) :** attribute is used to add delimiter or dump | for switching to new line.

Takes values true or false implying that text will

be newline

- **style:(optional)**

This attribute is used to provide styling to the highlight used.

■ **Syntax:**

```
<InputText locatorType="id" locatorExpression="Name" id="PartName"/>
```

So, this PartName is the test data csv column name and value of current row will be sent to web element.

Sample test data csv file:

"PartName", "PartDescription", "Policy"
"P000001242", "Engine Assembly", "EC Part"

P000001242 is value send to web element. Please check below image:

The screenshot shows the 'Create Part' dialog box. The 'Part Name' field is populated with 'P000001242' and has a yellow highlight around it. The 'AutoName' checkbox is empty. The 'Policy' dropdown is set to 'EC Part'. The background shows the 3DEXPERIENCE Platform interface with various navigation tabs and document lists.

Created with the Personal Edition of HelpNDoc: [Make the switch to CHM with HelpNDoc's hassle-free WinHelp HLP to CHM conversion tool](#)

InputWindowElement

InputWindowElement:

Clicks and inputs value in any element on the browser screen using an image of the element.

▪ **Syntax:**

```
<InputWindowElement imagepath="" skiperror="" id="" />
```

▪ **Attributes:**

- **imagepath:(mandatory)**

Specifies the path where the image of the screen element is located.

- **skiperror:(optional)**

Specifies whether to skip current test case or not. Values are true or false. Should be set to false to run test case. Default value is True.

- **id:(mandatory)**

This attribute is used to take value to search from csv. id attribute is already described in InputText tag. Please check it.

- **refid:(optional)**

The id attribute of FindElement tag will be used as refid for this tag.

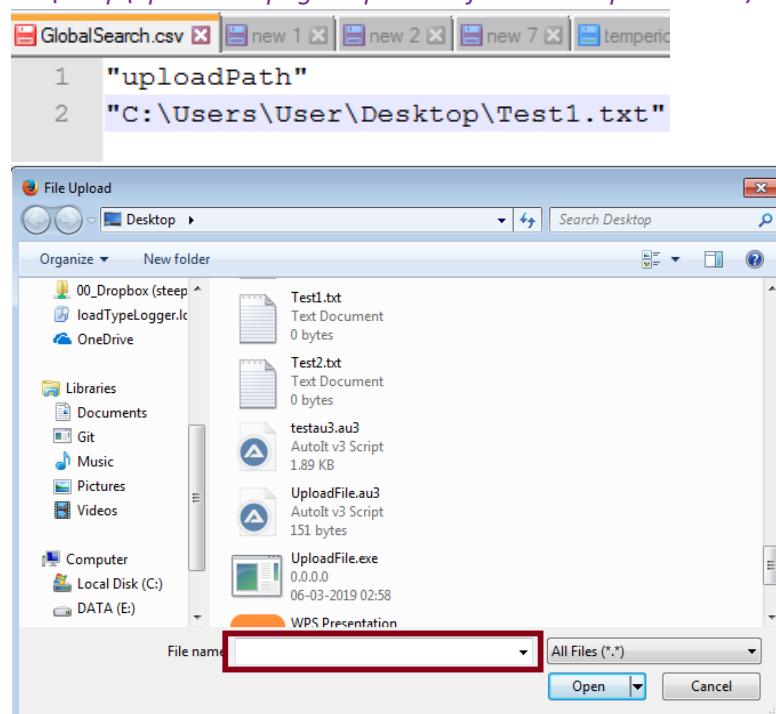
- **wait:(optional)**

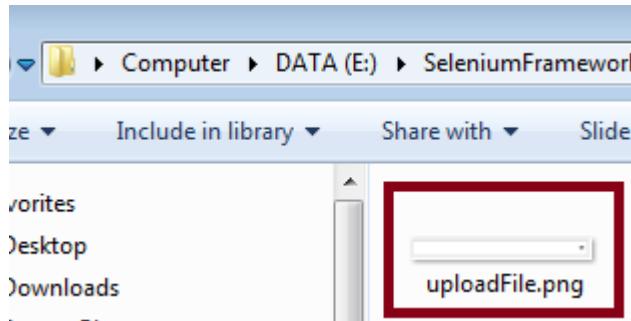
This attribute contains wait time in MS to find element. default wait will be taken from 3DX_TAS propertie's "3dx-tas.execution.step.timeout" key. Preferred wait is 10000..

▪ **Examples:**

In the image below, the user wants to enter file path in the file upload dialog. The tag in this case will be as follows:

```
<InputWindowElement imagepath="E:\SeleniumFramework\TestAutomationbuild-1.0\temp\uploadFile.png" skiperror="false" id="uploadPath"/>
```





Created with the Personal Edition of HelpNDoc: [Free EBook and documentation generator](#)

Lifecycle

Lifecycle:

Click on Promote/Demote lifecycle button.

- **Syntax:**

<Lifecycle action="" />

- **Attributes:**

- **action(mandatory):**

action attribute has two possible values

1. **promote:** clicks on promote button.
2. **demote:** clicks on demote button.

- **validateTargetState(optional):**

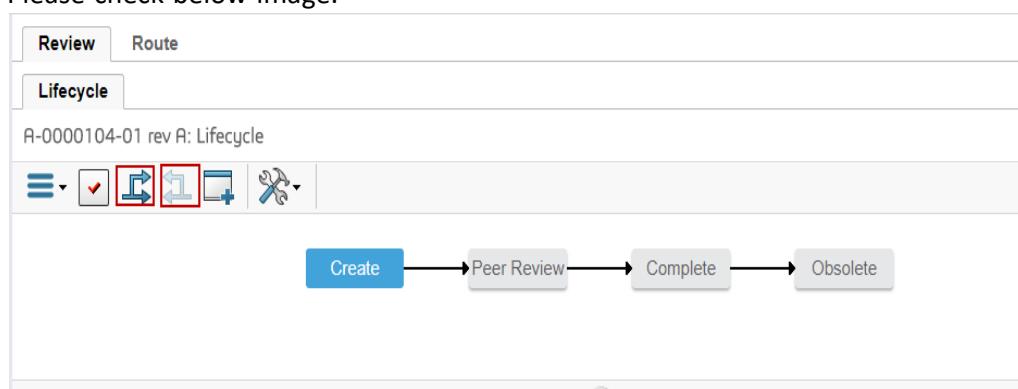
validateTargetState attribute has possible values as null, true or false

1. **true or null:** it will validate source and target state, both must not be same.
2. **false:** it will not validate the source and target state.

Before using this tag for promote/demote make sure script control is switch the frame on which lifecycle page is loaded.

- **Examples:**

Please check below image.



Created with the Personal Edition of HelpNDoc: [Free PDF documentation generator](#)

Login

Login:

With the help of this tag, user should be able to login to 3D space , by entering username, password .

- **Syntax:**

```
<Login username="$csv{username}" password="$csv{password}"/>
```

- **Attribute**

- ***username*(mandatory):**

This attribute refers to the valid username which needs to be entered in the username field, in order to login 3D space. This attribute must be passed in CSV file. This is compulsory attribute

- ***password*(mandatory):**

This attribute refers to the valid password which needs to be entered in the password field, in order to login to 3D space .This attribute must be passed in CSV file. This is compulsory attribute .

- ***closeWelcome*(Optional):**

this attribute is used to handle the welcome window after log in , possible values are true and false.

- ***location*(Optional):**

used to provide location value for Security Context.

- ***role*(Optional):**

used to provide role value for Security Context

- ***organization*(Optional):**

used to provide organization value for Security Context

- ***project*(Optional):**

used to provide project value for Security Context.

- **Examples:**

Enter user name and password for login in username and password attribute respectively.

Login tag will be used when we need to change user in between test case. Before using login tag, we need to logout from 3DEXperience.

```
<Login username="$csv{username}" password="$csv{password}"/>
```

If user wants to login into 3D space then this tag can be used .

3DEXPERIENCE ID

Email or username

Password

Remember me

Log in

Not registered? [Create your 3DEXPERIENCE ID](#)
[Forgot my password](#)

English

Created with the Personal Edition of HelpNDoc: [Don't Let Unauthorized Users View Your PDFs: Learn How to Set Passwords](#)

Logout

Logout:

With the help of this tag, user should be able to Logout from 3D space.

Attribute :

- **preCheck:(optional)**

This is optional tag which allow to precheck in any slideinwindow or openserchresult window before logging out.

- **isDashboard:(optional)**

Possible values are true or false.if value is true, then used to logout from 3DDashboard, else it is used to logout from 3dspace, default value is false.

- **wait:(optional)**

This attribute contains wait time in milliseconds to find element.Default value will be taken from 'sg-tas.execution.step.timeout'.Preferred wait is 2000MS.This attribute is only applicable for 2024xfd04.

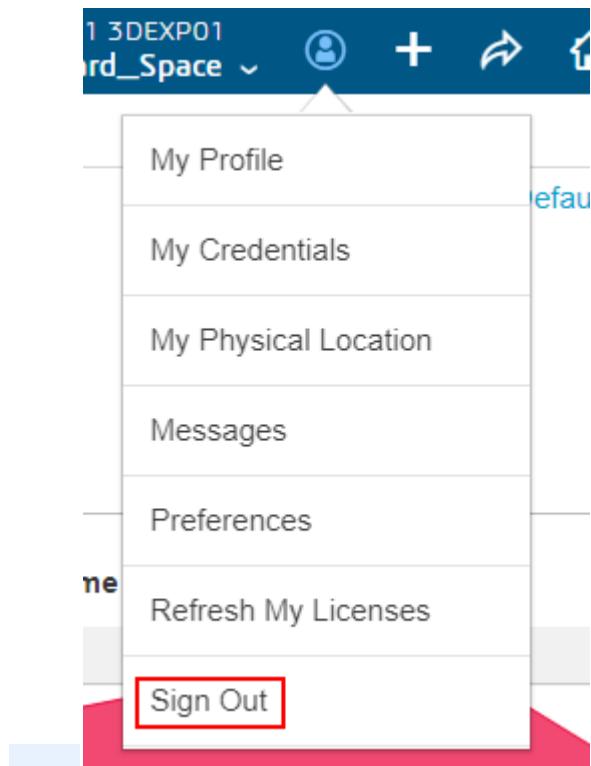
- **Syntax:**

<logout/>

- **Examples:**

if user wants to logout from 3D space then this tag can be used .

<logout/>



Created with the Personal Edition of HelpNDoc: [Easy to use tool to create HTML Help files and Help web sites](#)

MaximiseWindow

MaximiseWindow:

Maximise the current window. User can use this tag when they want to maximize the window. As shown in the below image user can maximize window

- **Syntax:**

```
<MaximiseWindow/>
```

A screenshot of a Microsoft Edge browser window titled "Create Product Line - Work - Microsoft Edge". The address bar shows a warning: "⚠ Not secure | https://www.steepgraph2021x.com/3dspace/common/emxCreate.jsp?type=type_ProductLine&typeChooser=false&autoNameChecked=false&showApply=true&submitAction=none&nameFile...". The page content is a form for creating a product line. It includes fields for Name, Type, Marketing Name, and Marketing Text. A note at the top says "Fields in red italics are required".

Created with the Personal Edition of HelpNDoc: [Easily create PDF Help documents](#)

MaximizeAndMinimize

MaximizeAndMinimize:

This tag is used to maximize or minimize a widget. Here, **appname** = Name of Application Widget, which needs to maximize or minimize.

If you want to maximize widget, give value = "True". If you want to minimize the widget, give value = "False".

▪ **Syntax:**

```
<MaximizeAndMinimize maximizeminimizeWidget="true"
appname="Requirements Structure Editor"/>
```

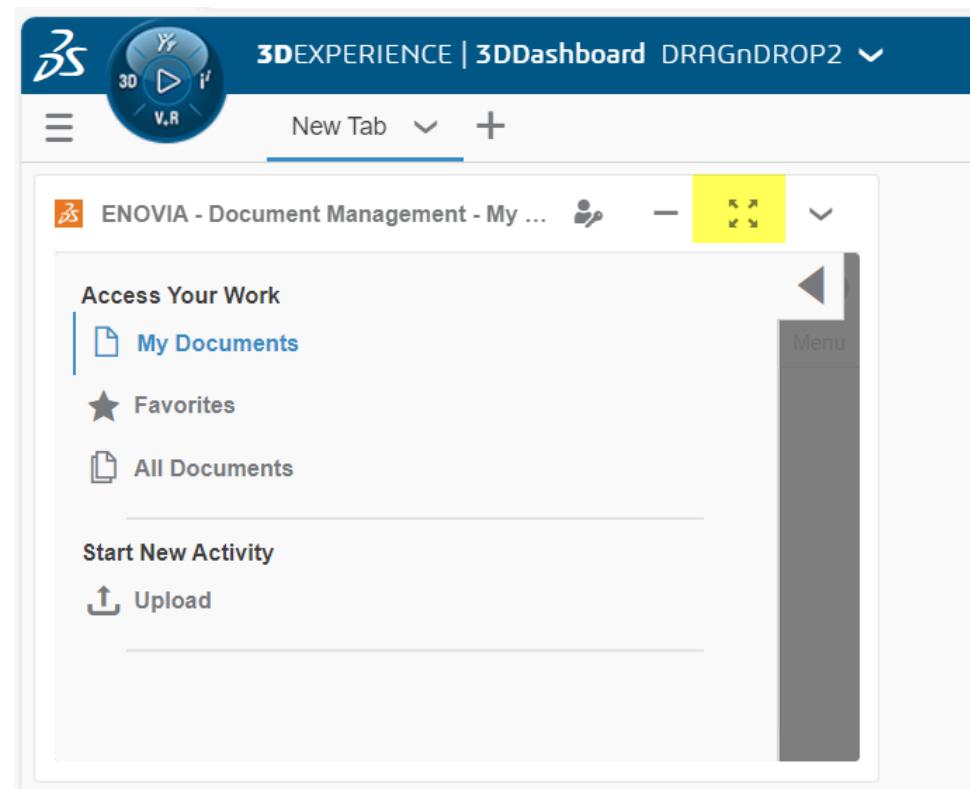
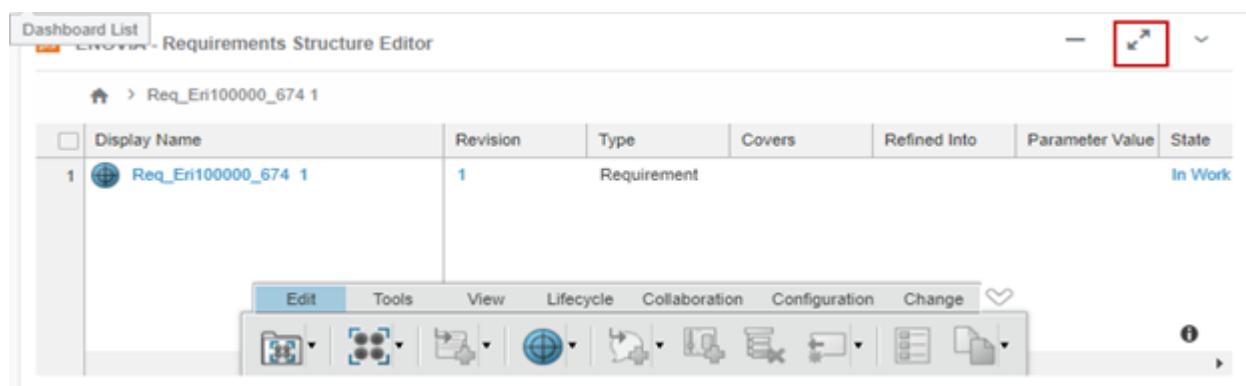
▪ **Attributes:**

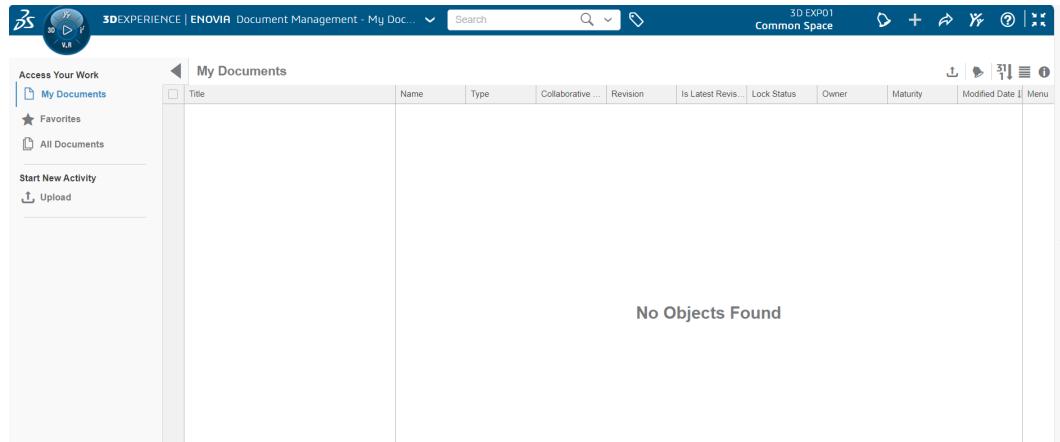
- **maximizeminimizeWidget(optional):**

If 'true', maximizes the widget, if 'false' minimizes the mentioned widget. Default value is false.

- **appname(mandatory):**

This attribute contains name of the widget to be maximized and minimized.





Created with the Personal Edition of HelpNDoc: [Elevate Your Help Documentation with a Help Authoring Tool](#)

MqlAssert:

MqlAssert:

Note: use RegisterObject tag before it to get the value of TNE or ObjectId. id/input are mandatory

Validate Test Case Execution against Enovia Server using MQL. before working on this tag, need to understand the MQL query first then go for test this tag :
Please make sure for understanding as below name's forms: NOTE : Please refer TAS-Prerequisite for MQLAssert tag.

TNR: Type | Name | Revision

ID: id means Object id of the product.

State(Lifecycle of objects): In Work: Preliminary, Frozen, Approved, Released, Obsolete.

What is attributes in MQL :

```
attribute[RepoPrivilege].value = 0
attribute[Originator] dump= admin_platform
attribute[StreamDescriptors].value =
attribute[PLMEntity.C_updatestamp].value = -1
attribute[PLMEntity.V_Name].value = Physical Product00000235
attribute[PLMEntity.V_isUptodate].value = 1
attribute[PLMEntity.PLM_ExternalID].value = prd-89686518-00000235
```

- **Syntax:** `<MqlAssert condition="validatetnr" id="" />`
- **Attributes:** This attribute specifies the condition to be validated in MqlAssert
 - **id : (Optional when input used) :**
id attribute is mapped to csv file column header. Input value should be taken from csv file.
 - **input : (Optional when id used) :**
input attribute is used to provide a hard-coded value for input instead of using id attribute from csv file.
 - **condition : (mandatory) :** There are 8 types of condition as below:
Provide the id values in CSV against which needs to be validated in id attribute
 1. validatetnr
 2. validateid
 3. validatestatefrommid
 4. validatestatefromtnr
 5. validateattributefromtnr
 6. validateattributefromid

7. validatedeletionfromid
8. validatedeletionfromtnr

- **selectExpression: (Optional):**
selectExpression attribute used when criteria is validateattributefromtnr, validateattributefromid to pass the selected expression in mql
- **errormessage : (optional)**
user can give hard coded message in this attribute .

Please check below Examples and Syntax for your references :

```

<Login username="$csv{username}" password="$csv{password}" />
<SwitchToDefaultContent />
Mention this value in CSV : Part|Part-0000100-01|A
<MqlAssert condition="validatetnr" id="test1"/>

Mention in CSV: object id 22513.60399.45900.2582
<MqlAssert condition="validateid" criteria="text" id="test1"/>

Mention this value in CSV : Part|Part-89686518-0001816-01|A|Preliminary
<MqlAssert id="tnr" condition="validatestatefromtnr"
selectExpression="attribute[V_Name] dump |" />

Mention this value in CSV : 22513.60399.45900.22528|Preliminary
<MqlAssert id="ValidatetStateFromIdCSV" condition="validatestatefromid" />

Mention this value in CSV : VPMReference|prd-89686518-00000235|A|FALSE
<MqlAssert id="tnr" condition="validateattributefromtnr"
selectExpression="select attribute[PLMReference.V_AuthorityControl] dump |"/>

Mention this value in CSV : 22513.60399.45900.22528|3DEXPO
<MqlAssert id="ValidatetStateFromIdCSV"
condition="validateattributefromid" selectExpression="select
attribute[Originator] dump |"/>

Mention this value in CSV :22513.60399.45900.22528
<MqlAssert condition="validatedeletionfromid"
id="ValidatetStateFromIdCSV" />

Mention this value in CSV : Part|Part-89686518-0001816-01|A
<MqlAssert condition="validatedeletionfromtnr" id="ValidatetStateFromIdCSV"
/>

```

How do you find Object id on UI(User Interface)

For Object ID : Please refere below :

on the property page pf the object , click on the edit button and then objectid can be found.

```

<Login password="$csv{password}"
username="$csv{username}"/>
<Wait time="6000"/>
<SwitchToDefaultContent/>
<ClickGlobalActionsMenu commandLabel="Engineering|Part|
Create Part"/>
<SwitchToSlideInWindow/>
<ClickElement
locatorExpression="//input[@name='autoNameCheck']" locatorType="xpath"/>
<InputText locatorExpression="VPMProductName1"

```

```

locatorType="id" input="TestTitle"/>
    <InputText locatorExpression="DescriptionId"
locatorType="id" input="DescriptionId"/>
    <ClickElement locatorExpression="//button[normalize-
space()='OK']" locatorType="xpath"/>
    <SwitchToDefaultContent/>
    <SwitchToFrame
name="content=>detailsDisplay=>portalDisplay=>ENCPartProperty"/>
    <ClickElement locatorType="xpath"
locatorExpression="//img[@src='..../common/images/iconActionEdit.png']"/>

    <RegisterObject
locatorExpression="(//div[@id='divPageBody'])//input[@name='objectId'][1]"
locatorType="xpath" id="Register_Model_Name" attribute="value" />
    <ClickElement locatorType="xpath"
locatorExpression="//button[contains(text(), 'Done')]"/>
    <SwitchToDefaultContent/>
    <SwitchToFrame
name="content=>detailsDisplay=>portalDisplay=>ENCPartProperty"/>

    <RegisterObject
locatorExpression="//tr[@id='calc_Type']//td[@class='field']"
locatorType="xpath" id="Type" removeSpace="true" />
    <RegisterObject
locatorExpression="//span[@class='object']" locatorType="xpath" id="Name"
removeSpace="true" />
    <RegisterObject
locatorExpression="//tr[@id='calc_Revision']//td[2]" locatorType="xpath"
id="Revision" removeSpace="true" />
    <SwitchToDefaultContent/>

    <MqlAssert condition="validateetstatefromtnr" id="aaa1"/>
    <MqlAssert condition="validateetattributefromtnr" id="aaa"
selectExpression="select attribute[Originator] dump"/>
    <MqlAssert condition="validatetnr" id="TNR" />
    <MqlAssert condition="validateetstatefromid" id="aaa2"/>
    <MqlAssert condition="validateetattributefromid" id="aaa3"
selectExpression="select attribute[Originator] dump"/>

```

Created with the Personal Edition of HelpNDoc: [Leave the tedious WinHelp HLP to CHM conversion process behind with HelpNDoc](#)

OpenActionToolbarMenu

OpenActionToolbarMenu:

Click the specified action toolbar menu command.

- **Syntax:**

<OpenActionToolbarMenu commandlabel="" />

- **Attributes:**

- **targetwindow(optional):**

Specifies the target location of the window after clicking the command eg. popup , content , slide in.

This attribute is applicable before Enovia version 22xfd02.

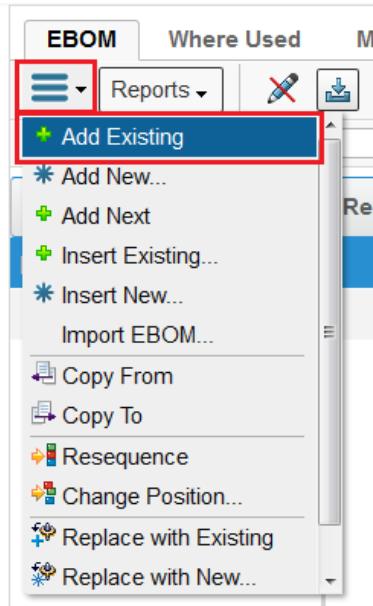
- **commandlabel(Mandatory):**

This attribute specifies the label of the command which will be clicked in UI.

- **Example:**

In the image below the script will click on the “Add Existing” menu command. For this the tag will be:

`<OpenActionToolbarMenu commandlabel="Add Existing"/>`



Note:

1. If the command is nested in a sub-menu in this case user will have to specify the commandlabel in a pipe separated fashion with the sub-menu label followed by the command label.
2. And OpenActionToolbarMenu used where web element name = "Action"

Created with the Personal Edition of HelpNDoc: [Full-featured Documentation generator](#)

OpenChooser

OpenChooser:

Click on the chooser button next to given field.

- **Syntax:**

`<OpenChooser fieldlabel="" />`

- **Attribute:**

- **fieldlabel(mandatory):**

The attribute fieldlabel will be used to find the chooser button. User must give label of field which has chooser button.

- **Example:**

Please check below image. Suppose we would like to assign Part Family. So, what we must click on chooser button next to that filed and search part family.

Similarly, we can use this tag to click on chooser button to open search form.

`OpenChooser fieldlabel="Part Family" />`

Create Part |



Fields in red italics are required

Type	Part	...
Part Name		<input type="checkbox"/> AutoName
AutoName Series	Not Selected	▼
Policy	Development Part	▼
Custom Revision Level	A	
Number of Parts	1	
Part Family		<input type="button"/> Clear

Created with the Personal Edition of HelpNDoc: [Streamline your documentation process with HelpNDoc's WinHelp HLP to CHM conversion feature](#)

OpenCompassApp

OpenCompassApp:

With the help of this tag, user could click any of the given Compass Quadrant and open the any of the required Compass App from it by using quadrant, appnameimage and appname attributes. User must provide quadrant, appnameimage and appname in order to open the application.

- **Syntax:**

```
<OpenCompassApp quadrant="" appname="">
```

NOTES:

1. The images captured for locatorImage attribute should always be in .png extension. The image captured for the required element should be focused and captured properly.
2. The quadrant, appnameimage and appname are mandatory attributes.

- **Attributes:**

- **quadrant:(optional)**

This attribute refers to the quadrant, from which user needs to launch the 3DX-TAS 2.0 188 / 252 required application. The possible values are north, south, east

and west. Attribute is used to open quadrant of compass App, Quadrant was mandatory field till 2020x.

- **wait:(optional)**

wait attribute is added for 21x for enhance performance issue. Default value will be taken from '3dx-tas.execution.step.timeout'. Preferred wait is 5000MS.

- **appIndex:(optional)**

if multiple app are returned by search , select desired app by its number in list viz 1, 2 etc.

- **appname:(mandatory)**

This attribute refers to the specific text of the appname, on which user needs to click, in order to launch the application.

▪ **Examples:**

In the image below, the user wants to open Product Structure App from north quadrant. The tag in this case will be as follows.North quadrant is same like our quadrant system.

```
<OpenCompassApp quadrant="north" appname="Variant Management"
appIndex="1"/>
```

- **note:**

To check the app index user need to inspect DOM.



Created with the Personal Edition of HelpNDoc: [Maximize Your Documentation Capabilities with HelpNDoc's Project Analyzer](#)

OpenNewTabInDashboard

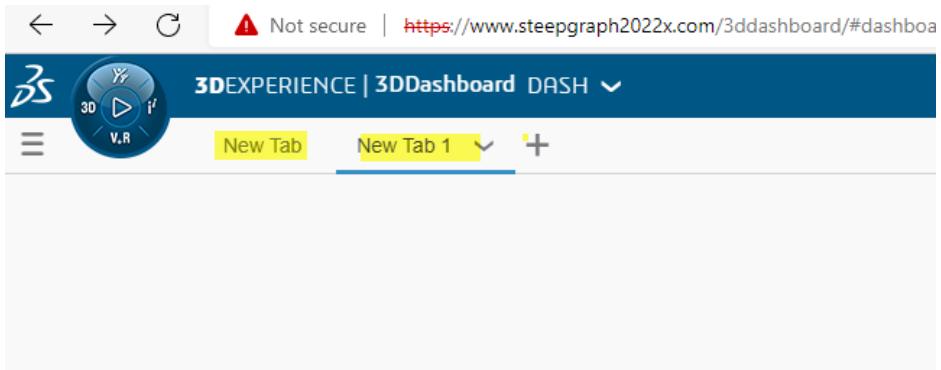
OpenNewTabInDashboard:

Used to create a new tab in dashboard. On creation of new tab, the pointer will move to the newly created tab.

▪ **Syntax:**

```
<OpenNewTabInDashboard/>
```

Examples:



Created with the Personal Edition of HelpNDoc: [Effortlessly upgrade your WinHelp HLP help files to CHM with HelpNDoc](#)

OpenNewToolbarInDashboard

OpenNewToolbarInDashboard :

With the help of this tag ,Open the new Tab in 3DDashBoard

- **Syntax:**

```
<OpenNewToolbarInDashboard toolbarName="" actionmenu="" dropdownOption="" />
```

NOTES:

1. Using this tag user can Create
2. In this tag user should write skipped based testcases.

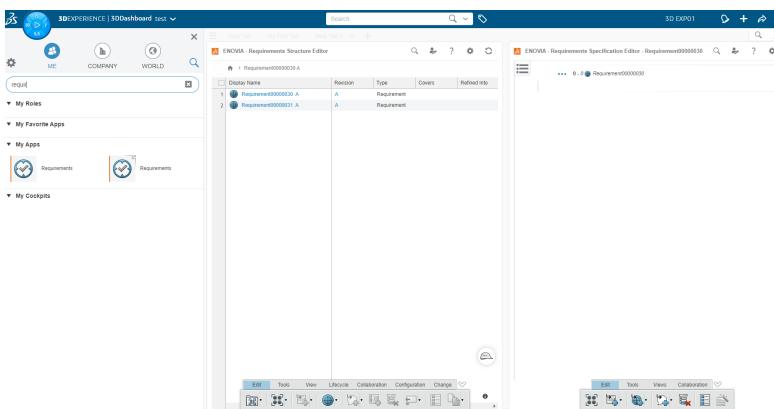
- **Attributes:**

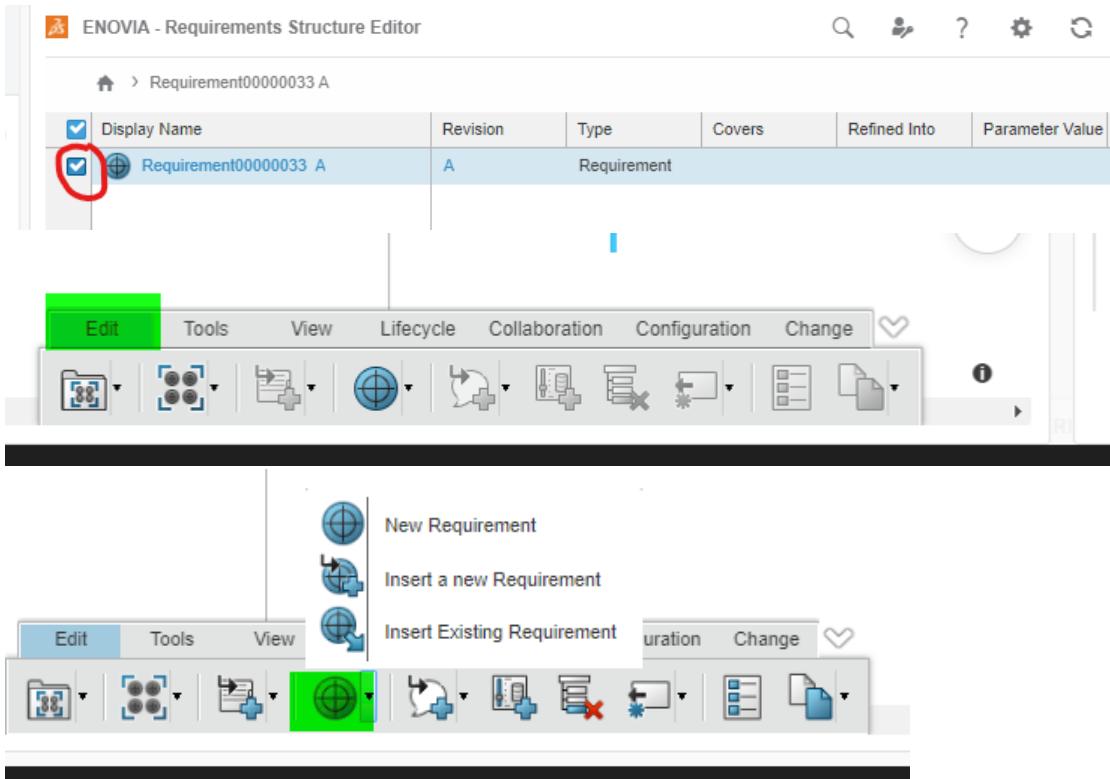
- **toolbarName:(Mandatory)**
name of Toolbar which user wants to select.
- **actionmenu:(Mandatory)**
Action on different action menu. Name of the menu in toolbar.
- **dropdownOption:(optional)**
Choose any dropdown option as per action menu name.

- **Examples:**

Suppose you want to write some skipped testcases.

```
<OpenNewToolbarInDashboard toolbarName="Edit" actionmenu="New Requirement" dropdownOption="New Requirement"/>
```





Created with the Personal Edition of HelpNDoc: [Effortlessly Edit and Export Markdown Documents](#)

OpenSearchResult

OpenSearchResult:

Open the object searched through global search. column and value attributes can be used to identify object based on some other column in the current view if there are multiple results with same name. If the search displays 0 results, then it clicks on clock to show results modified recently(results which were not indexed immediately).

If column or value is not used the both the attribute will be set to its default value. If customizedResult is true then column and value attribute will only applicable for non freezePane.

- **Syntax:**

```
<OpenSearchResult id="" criteria="" column="" value="" />
```

- **Attributes:**

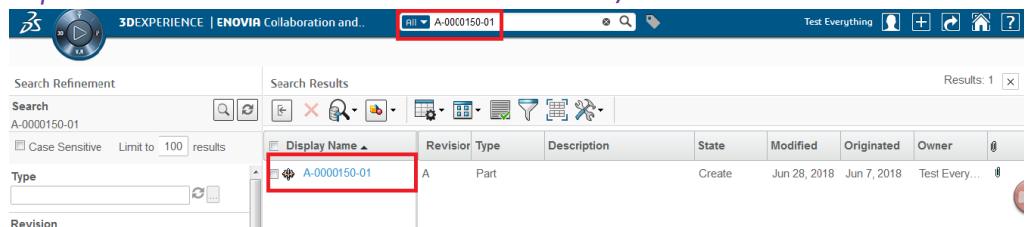
- **id(Option if input used):** id(CSV) attribute is mapped to CSV file column header, Input value should be taken from CSV file. This attribute will be the name column value.
- **input(Option if id used):** This attribute will be the name column value. input attribute is used to provide a hard-coded value for input instead of using id attribute from csv file.
- **criteria(optional):** possible values are 'text' or 'row-number', specifies the criteria on which the searched object will be opened. This attribute is only applicable before 3dx-tas.3dspace.release of V6R2019.
The possible values for this attribute are:
 1. text
 2. row-number

- **customizedResult: (optional)** This attribute is only for custom tables, only if the value is true, default value is false.
- **openWith : (optional)** : openWith attribute is for dashboard search result. User have to define the attribute value in this openWith result in which that to open that search result.
- **column: (optional customizedResult is false)** Name of column header to identify the object with same name.Possible values are Manufacturer ID,Manufacturer ASGID,Type,Title,,Description,Name,Revision,Owner,Creation Date,Modification Date. Default value is Name.If customizedResult is true then column attribute will only applicable for non freezePane.
- **value: (optional customizedResult is false)** Value of column header's value to identify the object .Default value is the value of id attribute.If customizedResult is true then value attribute will only applicable for non freezePane.
- **wait: (optional)** This attribute contains wait time in milliseconds to find element.Default value will be taken from '3dx-tas.execution.step.timeout'.Preferred wait is 3000MS.This attribute is only applicable for 21x , 22xfd02 and 22xfd06.

▪ **Example:**

In the image below object named “A-0000150-01” has been searched. To open it the following tag will be used:

`<OpenSearchResult id="SearchPart" criteria="text"/>`



In the csv file, the header “SearchPart” must exist for the object to be searched.

`SearchPart FilterPartByNameF
"A-0000150-01", "0001|0002", "`

`<OpenSearchResult id="OpenPart" criteria="text" column="Type" value="Product Line"/>`

Created with the Personal Edition of HelpNDoc: Streamline Your Documentation Creation with a Help Authoring Tool

OpenURL

OpenURL:

OpenURL tag let's user navigate to a given URL(within new tab or new window, default tab or default window). It has two attributes: url or id and target.

▪ **Syntax:**

`<OpenURL url="https://www.steepgraph.com/3dspace" target="self" />`

▪ **Attributes:**

- **url or id:(any one is mandatory)**

id is a csv column name and url is a direct/hard coded url. if url or id equals to null or empty then it will throw an exception.
if you want to give URL from 3dx-tas.tag.openurl.url key of browser property, then please use **3dx-tas.tag.openurl.url="https://www.google.com"**

- **target:(Optional)**

The possible values of this attribute are "tab" , "window" and "self":
tab: tab could be used when user wants to open url in new tab.
window: window would open url in new window.
self: self is used when user wants to open url in same window.

▪ **Example:**

```
<OpenURL url="https://www.steepgraphdev.com/3dpassport/cas/login?serverId&service=https%3A//www.steepgraphdev.com/3ddashboard/" target="self"/>
<Wait time="6000"/>
```

Created with the Personal Edition of HelpNDoc: [Enhance Your Documentation with HelpNDoc's Advanced Project Analyzer](#)

Option

Option :

With the help of this tag, Select single or multiple options from Select Box.

▪ **Syntax:**

```
<option selectList="" selectBoxId="" />
<option selectList="" selectBoxName="" />
```

NOTES:

1. This tag use to select the select list from select box
2. In this tag, Select Box Id & Select Box Name both are optional but anyone is mandatory.
3. In this tag user should give select list and any one between Select Box Id & Select Box Name.

▪ **Attributes:**

- **selectBoxId:(optional)**

This attribute is Option. ID of the selectbox from which options to be selected.

- **selectBoxName:(optional)**

This attribute is Option. Name of the selectbox from which options to be selected.

- **selectList:(mandatory)**

This attribute is Mandatory.1) For multiselect : Comma separated list of title of the options to be selected. eg. "AllowDelegation,Action"

2) For single select : Title of the option to be selected.

eg. "AllowDelegation"

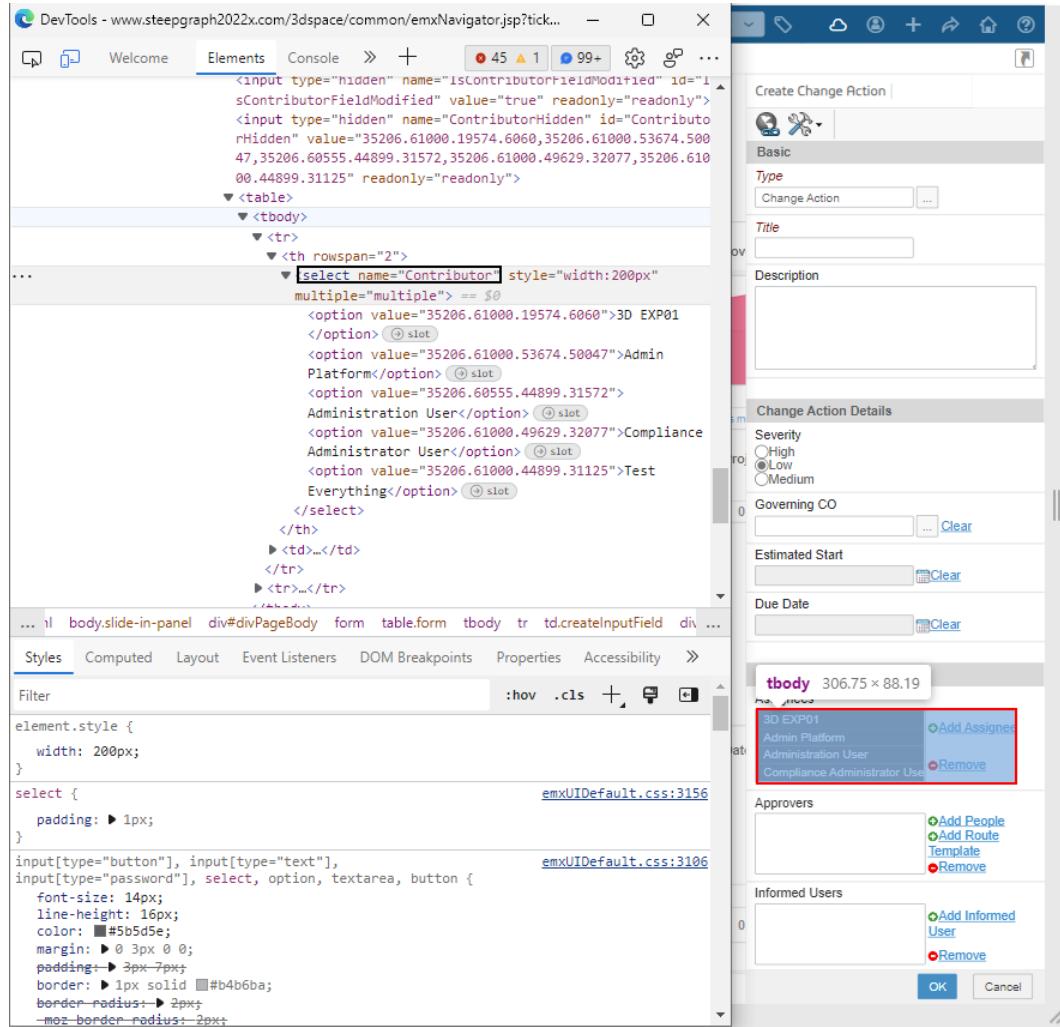
▪ **Examples:**

Suppose you want to select list from select box user has provide boxname by inspecting

DOM for that box. Also for BoxId user has to inspect DOM. In the option user has to give option separated by comma.

```
<option selectList="" selectBoxId="" />
```

```
<option selectList="3D EXP01,Test Everything,Administration User,Admin Platform"
selectBoxName="Contributor"/>
```



For SelectBoxName user has to inspect DOM.

```
<option selectList="3D EXP01,Test Everything,3D EXP02,Administration User,Admin Platform"
selectBoxName="Contributor"/>
```

The screenshot shows the 'Customize Table View' dialog in a browser. The 'Visible Columns' section is highlighted with a red box. Below the dialog, the browser's DevTools Elements tab is open, showing the HTML structure of the page. A specific select element under a td tag is highlighted with a red box. The styles panel on the right shows CSS rules for the select element.

Created with the Personal Edition of HelpNDoc: [Effortlessly Convert Your Word Doc to an eBook: A Step-by-Step Guide](#)

OptionalStart and OptionalClose

OptionalStart and OptionalClose :

OptionalStart is the Starting tag for tags which grouped between optionalStart and optionalClose and can be skipped even if it is failed

OptionalClose is the Ending tag for tags which grouped between optionalStart and optionalClose and can be skipped even if it is failed

User must need to use OptionalClose after using optionalStart.

- **Syntax:**

```
<optionalStart/>
<optionalClose/>
```

NOTES:

1. This tag use to Skipped some unwanted functionality
2. In this tag user should write skipped based test-cases.

- **Examples:**

Suppose you want to write some skipped test-cases

```
<Login username="$csv{username}" password="$csv{password}"/>
```

```

<optionalStart>
    <ClickGlobalActionsMenu commandLabel="" />
    <SwitchToDefaultContent />
    <ClickElement locatorType="xpath" locatorExpression="Testing" />
    <ClickGlobalActionsMenu commandLabel="Engineering/Part/Create Part" />
    <SwitchToSlideInWindow/>
    <InputText locatorType="id" locatorExpression="Name" input="InnovaCarsWheels19" />
    <ClickElement locatorType="xpath" locatorExpression="//button[text()='OK']" />
    <SwitchToDefaultContent />
    <SwitchToFrame name="content=>detailsDisplay=>portalDisplay" />
    <ClickElement locatorType="xpath" locatorExpression="//td[text()='Images']" />
    <ClickElement locatorType="xpath"
locatorExpression="//body/div[@id='topbar']/div[4]/div[1]/div[2]/div[1]/form[1]/div[1]/inputs[1]" />
        <ClickElement locatorType="xpath" locatorExpression="Testing" />
        <ClickElement locatorType="xpath" locatorExpression="xpath
locatorExpression="//body/div[@id='topbar']/div[4]/div[1]/div[2]/div[1]/form[1]/div[1]/inputs[1]" />
    <optionalClose/>

    <Wait time="3000" />
    <SwitchToDefaultContent />
    <SwitchToFrame name="content=>detailsDisplay=>portalDisplay" />
    <ClickElement locatorType="xpath" locatorExpression="//td[text()='Images']" />
    <ClickElement locatorType="xpath" locatorExpression="//td[text()='Revisions']" />
    <Wait time="3000" />

```

Created with the Personal Edition of HelpNDoc: [Make Your PDFs More Secure with Encryption and Password Protection](#)

OutlookLogin

OutlookLogin :

The OutlookLogin Tag is used to open email in Outlook. This tag cannot be applied to accounts where two-factor authentication is enabled.

- **Syntax:**

```
<OutlookLogin password="" username="" />
```

NOTES:

1. This tag use to login Outlook
2. In this tag user should give username and password
3. This tag cannot be applied to accounts where two-factor authentication is enabled.

- **Attributes:**

- **username:(Mandatory)**
this attribute provides username for login into Outlook.
- **password:(Mandatory)**
this attribute provides password for login into Outlook.
- **wait:(optional)**
This attribute contains wait time in milliseconds to find element. By default takes time from 3dx.tas.step.interval.

- **Examples:**

Suppose you want to login outlook login

```
<OutlookLogin username="$csv{username}" password="$csv{password}"
wait="$csv{if_wait}" />
```

```
<OutlookLogin username="admin123" password="admin123"/>
```

password	username	if_wait
OJ6uCS5JFB/Cv9ob92N5d9rgFsdhGrHplc9lFpJTUWg=	nitap@steepgraph.com	15000

Created with the Personal Edition of HelpNDoc: [Effortlessly Support Your Windows Applications with HelpNDoc's CHM Generation](#)

OutlookLogout

OutlookLogout :

With the help of this tag, Close the current 'self'(default), 'tab' and 'window'

- **Syntax:**

```
<OutlookLogout wait="" username="" />
```

NOTES:

1. This tag use to Logout Outlook

- **Attributes:**

- **username:(Mandatory)**

this attribute provides username for logout into Outlook.

- **wait:(Optional)**

This attribute is Optional. This attribute contains wait time in milliseconds to find element. Default value is 7000 ms.

- **Examples:**

Suppose User want to Logout the Outlook.

```
<OutlookLogout wait="2000" ussername="sheetalk@steepgraph.com"/>
```

Created with the Personal Edition of HelpNDoc: [Effortlessly optimize your documentation website for search engines](#)

Post & EndPost

Post & EndPost:

Post & EndPost:

1. POST tag indicate the **beginning of post process section of test case** and ENDPOST tag indicate the **end of post process section of test case**.

POST & ENDPOST Block tag should be **at the end of test script**. If Test case is getting failed it will skipped all tag after Failure and directly execute Post-Endpost tag.

The tags used between POST & ENDPOST tag will execute anyways.

2. User can use **POST & ENDPOST inside Included BB** also but, functionality remain same. User have to use **POST & ENDPOST tag Block** at the **end of Include Script** because it is the **part of Post processing Section**.

Inside Include **after POST & ENDPOST tag block no tag will be executed**.

For Example: As shown in the below image.

This is Main Script

```

1  ../../resources/xsd/TestAutomationFramework.xsd (xsi:schemaLocation)
2  <?xml version="1.0" encoding="UTF-8" standalone="no" ?>
3  <TestCase xmlns="https://www.steepgraph.com" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4    xsi:schemaLocation="https://www.steepgraph.com ../../resources/xsd/TestAutomationFramework.xsd">
5    <Login username="$csv{username}" password="$csv{password}" />
6    <SwitchToDefaultContent />
7    <Include filepath="testsuites\BB\Create.xml" />
8    <Wait time="300"/>
9    <SwitchToDefaultContent/>
10   <Post/>
11   <Wait time="2000"/>
12   <ClickGlobalActionsMenu commandLabel="Engineering/Part/Create Part"/>
13   <Include filepath="testsuites\BB\CreatePartBB.xml"/>
14   <EndPost/>
15
16
17
18 </TestCase>

```

This is include BB Script

```

1  ../../resources/xsd/TestAutomationFramework.xsd (xsi:schemaLocation)
2  <?xml version="1.0" encoding="UTF-8" standalone="no" ?>
3  <TestCase xmlns="https://www.steepgraph.com" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4    xsi:schemaLocation="https://www.steepgraph.com ../../resources/xsd/TestAutomationFramework.xsd">
5
6    <Wait time="2000"/>
7    <ClickGlobalActionsMenu commandLabel="Engineering/Part/Create Part"/>
8    <SwitchToSlideInWindow/>
9    <ClickElement locatorExpression="//table[@class='form']//input[@name='autoNameCheck']" locatorType="xpath"/>
10   <InputText locatorExpression="DescriptionId" locatorType="id" input="Part"/>
11   <HighLightElement locatorExpression="DescriptionId" locatorType="id" id="DescriptionId"/>
12   <ClickElement locatorExpression="//button[@class='btn-primary']" locatorType="xpath"/>
13
14   <Post/>
15   <Wait time="2000"/>
16   <SwitchToDefaultContent/>
17   <EndPost/>
18
19 </TestCase>

```

For Pass Cases(Post/EndPost Behavior): All the tag should be executed inside Included BB as well as in Main Script executed with POST & ENDPOST block.

After the execution of POST & ENDPOST Block inside the Include BB.

The Script return back to Main Script It should Executed that other tag given in the main script and then Executed the POST & ENDPOST tag Block.

For Fail Cases (Post/EndPost Behavior): For Fail cases Two Scenarios

1. Tag Fail In Main Script: If the execution Fail in Main Script it should only Executed POST & ENDPOST block in Main Script.

2. Tag Fail inside Include BB:

If any tag fail inside the execution of include BB, it should executed the POST & ENDPOST block inside

include BB after execution return back into Main Script and Executed the POST & ENDPOST block in Main Script In fail cases the error message is present in log for Skipped Tag, before the POST/ENDPOST block execution. As shown in the below image.

```

2024-01-03 13:20:33,619 [3DX-TAS] - Start of testCasePostProcessing
2024-01-03 13:20:33,619 [3DX-TAS] - Wait Tag processing is skipped because of script failure
2024-01-03 13:20:33,619 [3DX-TAS] - SwitchToDefaultContent Tag processing is skipped because of script failure
2024-01-03 13:20:33,619 [3DX-TAS] - Include.xml::Line-> 15 => TestPlan => Include => <Post> Processing
2024-01-03 13:20:33,620 [3DX-TAS] - Starts of Post/EndPost

```

Note : The test suites **execution's result is not dependent on POST & ENDPOST TAG.**
 POST & ENDPOST tag will be used once at the End of any XML Script. After
ENDPOST No tag will be executed as, it is the Functional behavior of the tag.

- **Syntax:**

```
<Post />
<EndPost />
```

- **Example:**

In below you can find the example

```
i) <Login password="$csv{password}" username="$csv{username}"/>
   <ClickGlobalActionsMenu commandLabel="Project/Create Project/New"/>
   <ClickElement locatorType="xpath"
locatorExpression="//input[@name='autoNameCheck']"/>
   <ClickElement locatorType="xpath" locatorExpression="//button[@type='button']"/>
   <SwitchtoDefaultContent/>
<Post/>
   <Include filepath="testsuites/TAG WISE TEST CASE/BBT/include.xml"/>
<EndPost/>
```

In the next section the script will fail in last clickelement tag due to wrong xpath.

```
ii)<Login password="$csv{password}" username="$csv{username}"/>
   <ClickGlobalActionsMenu commandLabel="Project/Create Project/New"/>
   <ClickElement locatorType="xpath"
locatorExpression="//input[@name='autoNameCheck']"/>
   <ClickElement locatorType="xpath"
locatorExpression="//buttdxgon[@type='button']"/>
<Post/>
   <Include filepath="testsuites/TAG WISE TEST CASE/BBT/include.xml"/>
<EndPost/>
```

Created with the Personal Edition of HelpNDoc: [Free help authoring environment](#)

PinWidgetToDashboard

PinWidgetToDashboard:

PinWidgetToDashboard tag let's user to pin a widget to a dashboard.

- **Syntax:**

```
<PinWidgetToDashboard />
```

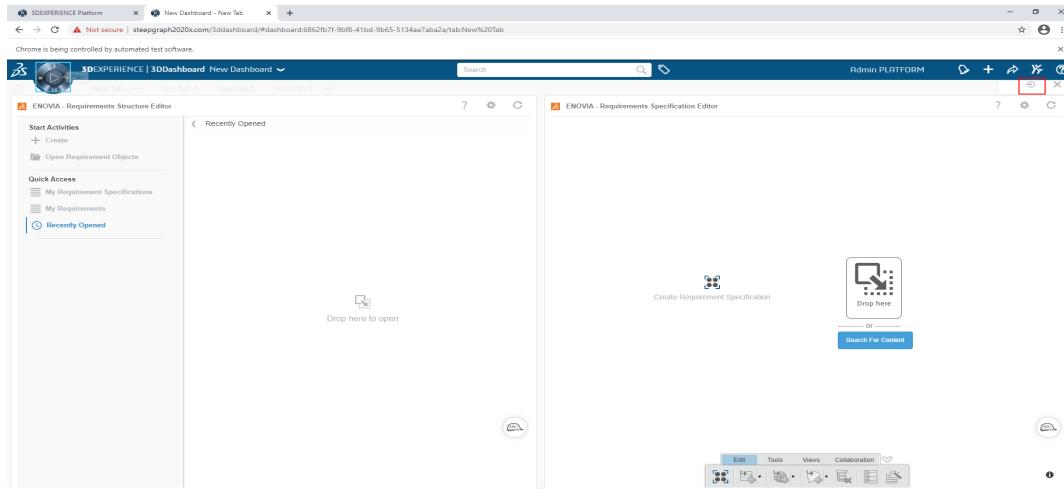
- **Attributes:**

- **pinBtnInWidget:(optional)**

possible values are 'true' or 'false'. Default value is 'false'.

Used to differentiate cases where pin button is inside widget its value will be false and where pin button is in header its value will be true for PinWingetToDashboard

<PinWidgetToDashboard PinBtnInWidget ="true" />



PinWidgetToDashboard value will be false when pin button will be inside the widget. As shown in the below image.

<PinWidgetToDashboard PinBtnInWidget ="false" />



Created with the Personal Edition of HelpNDoc: [Step-by-Step Guide: How to Turn Your Word Document into an eBook](#)

Print

Print:

Print message on console and in debug log file.

- **Syntax:**

<Print message="" />

- **Attribute:**

- **message(Mandatory):** This attribute specifies the expression you want to print.

The possible values for this attribute are:

1. \$currentFrame: This will print the name of the current frame in which the script control is.
2. \$currentWindow: This will print the name of the current window in which the script control is.
3. \$lineNumber - it will print line number of current executing line in the log as well as in console file
4. \$currentFileName - it will print current executing file name in the log as well as in console file
5. \$currentDate - it will print system date in the log as well as in console file
6. \$currentStep - it will print Step tag id and title in the log as well as

in console file

- **Example:**

In the image below, the debug log has printed the current frame's name. The frame name is also printed on the console. The tag syntax for this case is:

<Print message="\$currentFrame"/>

```
[Ljava.lang.Object;@13e344d
TestCaseSample2.xml => Suite => Test2 => Processing
TestCaseSample2.xml => Suite => Test2 => <GlobalSearch> Processing
TestCaseSample2.xml => Suite => Test2 => <OpenSearchResult> Processing
TestCaseSample2.xml => Suite => Test2 => <ClickCategoryCommand> Processing
TestCaseSample2.xml => Suite => Test2 => <SwitchToPortalDisplayFrame> Processing
TestCaseSample2.xml => Suite => Test2 => <SwitchToFrame> Processing
TestCaseSample2.xml => Suite => Test2 => <Print> Processing
$currentFrame : ENCBOM
TestCaseSample2.xml => Suite => Test2 => <FindElement> Processing
TestCaseSample2.xml => Suite => Test2 => <Action> Processing
TestCaseSample2.xml => Suite => Test2 => <SelectIndentedTableRow> Processing
```

In the image below, the debug log has printed the current window name. The frame name is also printed on the console. The tag syntax for this case is:

<Print message="\$currentWindow"/>

```
364 2022-09-30 11:40:08,171 [3DX-TAS] - End of parseCSVAttribute
365 2022-09-30 11:40:08,171 [3DX-TAS] - Start of parseREGAttribute
366 2022-09-30 11:40:08,172 [3DX-TAS] - End of parseREGAttribute
367 2022-09-30 11:40:08,172 [3DX-TAS] - Print.xml => TestPlan => Print => <Print> Processing
368 2022-09-30 11:40:08,172 [3DX-TAS] - attributeMap: {expression=$currentWindow}
369 2022-09-30 11:40:08,172 [3DX-TAS] - strExpression: $currentWindow
370 2022-09-30 11:40:08,175 [3DX-TAS] - $currentWindow : 3DEXPERIENCE Platform
371 2022-09-30 11:40:08,179 [3DX-TAS] - Started onTestSuccess
372 2022-09-30 11:40:08,179 [3DX-TAS] - Start of formatDate
```



Created with the Personal Edition of HelpNDoc: [Free Web Help generator](#)

ResetSearch

ApplySearch:

User can use **ApplySearch** tag when use choosing advance search type then this tag will used for reset the Search filter form.

- **Syntax:**

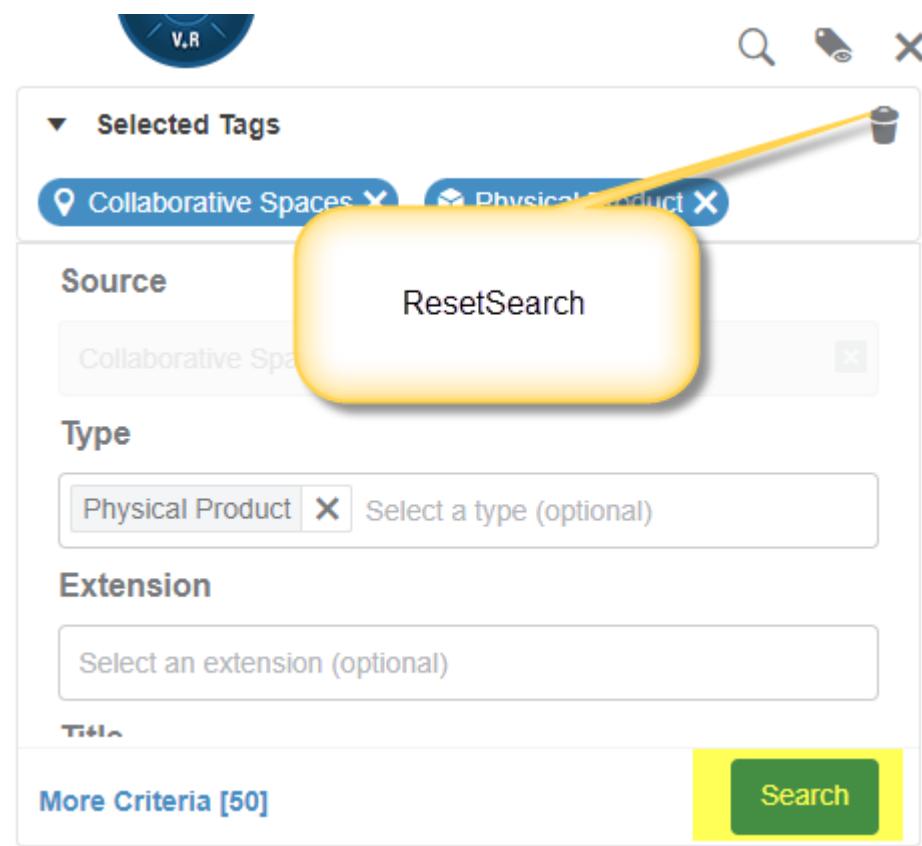
<ResetSearch/>

- **Example:**

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<TestCase xmlns="https://www.steepgraph.com" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="https://www.steepgraph.com ../../resources/xsd/TestAutomationFramework.xsd">

<Login username="$csv{Username}" password="$csv{Password}" />
<ClickElement locatorExpression="//div[@id='searchFieldDropdown']" locatorType="xpath"/>
<ClickElement locatorExpression="//li[@title='Advanced Search in the 6WTagger']" locatorType="xpath"/>
<Wait time="5000"/>
<InputText locatorType="xpath" locatorExpression="//input[@placeholder='Select a type (optional)']" input="Physical product"/>
<ClickElement locatorType="xpath" locatorExpression="//span[text()='Physical Product']" />

<ApplySearch/>
<Wait time="5000"/>
<ResetSearch/>
<Wait time="5000"/>
<Logout/>
</TestCase>
```



Created with the Personal Edition of HelpNDoc: [Free PDF documentation generator](#)

ReadUserInput (Deprecated)

ReadUserInput (DEPRECATED):

This requirement is invalid as RegisterObject was developed to have a dynamic value that the target application generates

- **Syntax:**

```
<ReadUserInput/>
```

NOTES:

1. This tag use to only when CSV is download in downloaded folder due to running time then it will work.

Created with the Personal Edition of HelpNDoc: [News and information about help authoring tools and software](#)

ReadXML

ReadXML :

With the help of this tag, ReadXML another test case to UnZip in current zipfile from browser Location or manually location.

- **Syntax:**

```
<ReadXML unzip="true" fromBrowserDownloadLocation="true" id="" deleteFile="true" downloadFileExt="" regexExpr="" unziplocation="" />
```

NOTES:

1. This tag use to only when CSV is download in downloaded folder due to running time

- then it will work.
2. In this tag user should give proper file path you want to validate into the CSV file with Extension.
 - **Attributes:**
 - ***id:(Optional if input used)***
Path of the building block (zip file) present in testsuites.the id attribute of ReadXML tag is used as refid for AssertXML tag.
 - ***input:(Optional if id used)***
Path of the building block (zip file) present in testsuites.the id attribute of ReadXML tag is used as refid for AssertXML tag.
 - ***fromBrowserDownloadLocation:(Mandatory)***
Path of the building block (zip file) present in testsuites.
 - ***unZipLocation:(Mandatory)***
Path/Dir of the zip file where it has been extracted. unzip file folder..
 - ***unzip:(Mandatory)***
Boolean Check for file to unZip.
 - ***downloadFileExt:(Mandatory)***
Provide extension of file of Unzip File i.e xml/pdf/word
 - ***regexExpr:(optional)***
Provide regexExpr only when xml file contains regular expression.
 - ***deleteFile:(optional)***
Default value is true, true or null, false value for delete and undelete zip and xml file respectively.

▪ **Examples:**

Suppose you want to validate header list from CSV file.

```
<ReadXML unziplocation="testsuites\2021x_ERIMATE_Replacement\UNZIPPED_LOCATION" unzip="true"
id="downloadedFile" fromBrowserDownloadLocation="true" regexExpr="true" downloadFileExt="zip"
deleteFile="true"/>

<ReadXML unziplocation="C:\Users\SGSPC044\Downloads\PinWidgetToDashboard\
unzip="true" id="downloadedFile fromBrowserDownloadLocation="true" regexExpr="true"
downloadFileExt="zip" deleteFile="false"/>
```

Created with the Personal Edition of HelpNDoc: [Effortlessly bring your documentation online with HelpNDoc](#)

RegisterIndentedTableObjects

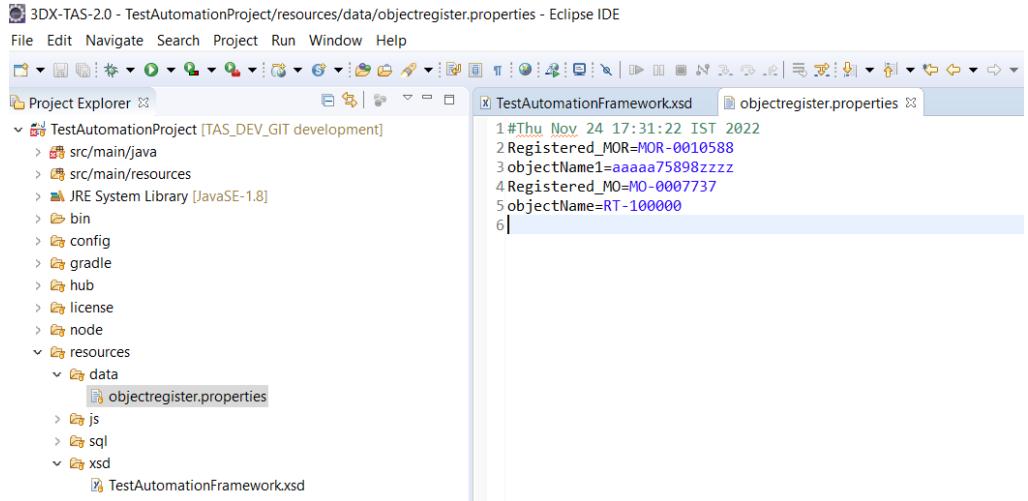
RegisterIndentedTableObjects:

User can use this Tag to register the current indented table row **After creation**.

This tag should be used after creating new objects and after use of
SelectOrOpenNewIndentedtableRow tag.

- **Syntax:**
`<registerindentedtableobjects/>`

```
<SelectIndentedTableRow expand="true" id="SelectRow" criteria="level"/>
<RegisterIndentedTableObjects/>
```



```
<RegisterIndentedTableObjects/>
<ClickElement locatorType="xpath"
locatorExpression="//td[@title='Product Package']//img[1]"/>
<ClickElement locatorType="xpath" locatorExpression="//label[text() ='Create Product Package']"/>
<SwitchToDefaultContent/>
<SwitchToSlideInWindow/>
<CheckElement locatorType="xpath"
locatorExpression="//tr[@id='calc_Revision']//td[@class='createInputField']" criteria="visible"/>
<InputText locatorType="id" locatorExpression="MarketingName"
id="PP Designation"/>
<InputText locatorType="id" locatorExpression="MarketingTextId"
id="Marketing Text"/>
<InputText locatorType="id" locatorExpression="DescriptionId"
id="Description"/>
<Wait time="2000"/>
<!-- Click OK -->
<ClickElement locatorType="xpath"
locatorExpression="//button[@class='btn-primary']"/>
<SwitchToDefaultContent/>
<SwitchToContentFrame/>
<Wait time="5000"/>
<SelectOrOpenNewIndentedTableRow criteria="select"/>
```

RegisterObject

RegisterObject:

Register an object's name in a property file or in database by locatorType and locatorExpression for future reference.

To determine the destination in which the object's name will be registered, there must exist in TestAutomationFramework.properties file the following keys:

- **object.registration.type:** Specifies where to register the object name. It can take two values, either property file or database.
E.g. object.registration.type=propertyfile
- **object.registration.filepath:** Specifies the name of the file and path in your file system where it will be located.
E.g. object.registration.filepath=src\\main\\resources\\objectregister.properties.

▪ Syntax:

```
<RegisterObject locatorType="" locatorExpression="" id="" removeSpace="" />
```

▪ Attribute:

- **Locator Type: (mandatory)**

There are 4 types of locator: id, cssselector, name and xpath which are already discussed in ClickElement tag.

- **locatorExpression:(mandatory)**

This attribute contains expression specific to locator type.

- **id:(optional if input used)**

id attribute is mapped to csv file column header .id attribute column value is used to register the value in register properties with the column value in CSV file.

Please check it.if it is id equals to blank then it takes empty key in objectregister.properties

- **inout:(optional if id used)**

attribute is used to provide a hard-coded value for input instead of using id attribute from csv file

- **attribute:(optional)**

It specifies the attribute available on web elements to be register.E.g.- name, value, class, rmbldm, rmbrow etc.

- **removeSpace:(optional)**

If set to 'true', it removes all spaces in the value to be registered. Default value is 'false'.

- **length:(optional)**

It is used for prefix value **before** the objects.

- **prefix:(optional)**

It is used for prefix value **before** the objects.

- **suffix:(optional)**

It is used for prefix value **after** the objects'.

- **numeric:(optional)**

It is used for auto generation of numeric values. The attribute have Boolean value **true or false**. It Default value is false. If user want to generate number then set the value true. If actual length is < 1 then, numeric value cannot be generated and if actual length = 1 and with numeric value is true and user is using prefix and suffix with it then it will generated 0 every time and without prefix and suffix if the actual length =1 then it will generated numeric value between 0 to 9 .

For Example : User need to give numeric = true and the length for generating the number. It auto generate the number and register that number in the **ObjectRegister.Property** File. As shown in the below image.

```
<RegisterObject id="Register_Name1" length="4" numeric="true"/>
<InputText locatorType="xpath" locatorExpression="//input[@id='Name']" id="Name1" />
```

Create Part Specification

Type: Part Specification

Name: 1129 AutoName

AutoName Series: A Series

```
1 #Thu Aug 24 12:07:36 IST 2023
2 Registered_Name1=1129
3 i=9
4 g=7
5 f=6
6 e=5
7 d=4
8 b=2
```

	A	B	C	D	E
1	password	username	Register_Name1	Name1	
2	T0g3aShU6v47E1iaY3JEaw==	3DEXPO1	Registered_Name1	\$reg{Registered_Name1}	
3					

- **Example:**

In the images below, the user wants to register the name of the Part in a property file. The tag in this case will be:

```
<RegisterObject locatorType="xpath"
locatorExpression="//div[@id='divExtendedHeaderContent']//div[@id='divExtendedHeaderName']/span[1]" id="Part_Name"/>
```

The screenshot shows a product card for 'A-0000150-01'. The card includes fields for State (Create), Owner (Test Everything), and Modified (6/28/2018 6:54:11 PM). Below the card are tabs for 'Properties', 'Images', and 'Where Used'. The 'Properties' tab shows the name 'A-0000150-01' and other properties like 'Bill Of Materials' and 'Equivalents'. A red box highlights the product number 'A-0000150-01'.

In the csv file, a header with the name **Part_Name** must exist whose value will be the key with which the object will be registered.

`objectregister.properties`

```
1 Registered_Part=A-0000150-01
```

In Register object user can use new value to validated current product creation date with current system data, or print current system date. The default date format for taking system date is this **dd/MM/yyyy** means it will display date in this way **01/11/2023**. User can change this date format as per their requirement. According to that format it will display the System date.

The Syntax for getting system date in any xpath is : `$reg{currentDate}`. User can use this Syntax in any xpath where the want to validate. But user have to **define date format** as per the validation.

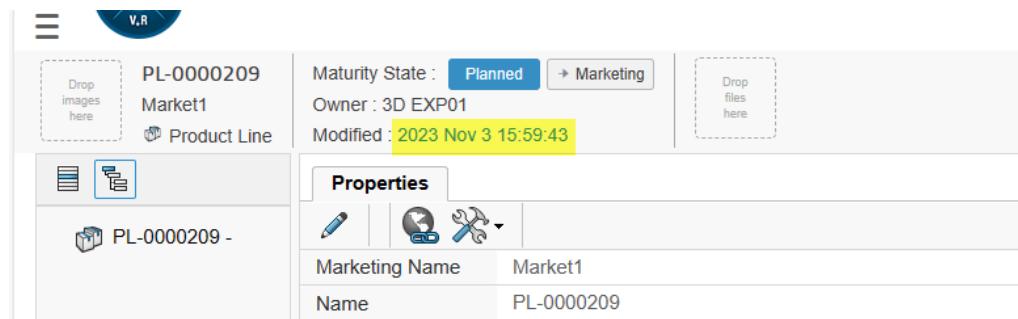
User can define the date format in this value itself or in browser Properties. If the user **cannot define any date format** then it **will take default format** as shown above.

If user have to define date format in value that is in xpath then the syntax is : `'$reg{currentDate####}'`.

The browser Property key name where user can define the value for date format is as below.

`3dx-tas.browser.date.format=MM/dd/yyyy`

For Example: If user have to check whether this Product is created Today. I have to validated this date as shown in the below image.



User have to first take out the xpath of that place.

The screenshot shows the browser's developer tools (Elements tab) with the following XPath selected:

```
//span/a[contains(text(),'2023 Nov 3')]
```

The xpath for this date is : //span/a[contains(text(),'2023 Nov 3')]. User have to replace this date '2023 Nov 3' with this expression
`'$reg{currentDate##yyyy MMM d##}'`.

So, the user have to use this xpath to validated date.As shown below.

```
<CheckElement locatorType="xpath"
locatorExpression="//span/a[contains(text(), '$reg{currentDate##yyyy MMM d##}')]" criteria="visible"/>
```

At the run time where user have given use this attribute it will automatically take system current date. As shown in below log image.

```
237 2023-11-03 12:15:25,235 [3DX-TAS] - End of parseCSVAttribute
238 2023-11-03 12:15:25,236 [3DX-TAS] - Start of parseEGAttribute
239 2023-11-03 12:15:25,236 [3DX-TAS] - Start of parseEGToolVariables
240 2023-11-03 12:15:25,236 [3DX-TAS] - Matched String : $reg{currentDate##yyyy MMM d##}
241 2023-11-03 12:15:25,236 [3DX-TAS] - End of parseEGToolVariables
242 2023-11-03 12:15:25,236 [3DX-TAS] - End of parseEGAttribute
243 2023-11-03 12:15:25,236 [3DX-TAS] - ClickRefreshButton.xml::Line-> 27 => TestPlan => ClickRefreshButton => <CheckElement> Processing
244 2023-11-03 12:15:25,236 [3DX-TAS] - attributeMap: {criteria=visible, locatorType=xpath, locatorExpression//span/a[contains(text(),'2023 Nov 3'))]
245 2023-11-03 12:15:25,236 [3DX-TAS] - Start of checkElement
246 2023-11-03 12:15:25,236 [3DX-TAS] - locatorType: xpath
247 2023-11-03 12:15:25,236 [3DX-TAS] - locatorExpression: //span/a[contains(text(),'2023 Nov 3'))]
248 2023-11-03 12:15:25,236 [3DX-TAS] - strCriteria: visible
249 2023-11-03 12:15:25,236 [3DX-TAS] - strResult: 30d
250 2023-11-03 12:15:25,358 [3DX-TAS] - aria-readyonly : null
251 2023-11-03 12:15:25,368 [3DX-TAS] - Assertion criteria passed: Specified element is visible.
```

Created with the Personal Edition of HelpNDoc: [Revolutionize your documentation process with HelpNDoc's online capabilities](#)

RunMQL

RunMQL :

With the help of this tag, Run the given MQL and validate the result

- **Syntax:**

```
<RunMQL id="" condition="" errormessage=" "/>
```

NOTES:

1. In this tag user set the ID attribute which is mapped with the csv.
2. user set the condition for specific cases.
3. user can pass the error message.

- **Attributes:**

- ***Id:(optional when input used)***
id attribute is mapped to csv file column header. Input value should be taken from csv file.
- ***input:(optional when id used)***
This attribute is used to provide a hard-coded value for input instead of using id attribute from csv file.
- ***condition:(optional)***
There are nine types of condition: none,=,!,<,>,<=,>=, contains,not contains. Default value is none.
- ***errorMessage:(optional)***
User can give hardcoded message in this attribute.
- ***result:(optional)***
this attribute is used to take the value of expected output of the given mql query, and this attribute is applicable when the condition attribute's value is not none.

- **Examples:**

Suppose you want to write some skipped testcases.

```
<RunMQL id="mql" condition="=" errormessage="this is an errormessage" />
<RunMQL input="print bus 'Part' 'Part-87748758-0000810-01' 'A' select
attribute[Annual Number of EU Units Imported] dump" condition="=" result="10" />
```

Created with the Personal Edition of HelpNDoc: [Effortlessly Spot and Fix Problems in Your Documentation with HelpNDoc's Project Analyzer](#)

Step & EndStep

Step & EndStep:

User use **Step** tag to create a step block which contains certain functional & end the clock using **EndStep** tag.

- **Attribute**

- **id: (mandatory)**
used to give unique id for group of steps
- **title: (mandatory)**
used to give unique name for group of steps

- **Syntax:**

```
<Step id="" title="" />
<EndStep/>
```

- **Example:**

If user wants to group Create Part step, following tag should be used.
to close the current step user can use EndStep tag.

```
<Step id="step1" title="Create Part"/>
<Click.... />
    <SwitchToSlidIn.... />
<Input..../>
<EndStep/>
```

Created with the Personal Edition of HelpNDoc: [Free iPhone documentation generator](#)

ScrollToElement

ScrollToElement:

Move scroll bar to make element visible.

- **Syntax:**

```
<ScrollToElement locatorType="id" locatorExpression="" />
```

- **Attribute:**

[LocatorType](#) and [LocatorExpression](#) is already discussed in ClickElement tag.

Sometimes web element on which user want to perform action is not visible on screen. If we directly tried to perform some action like click, then it will fail. For such cases user need to use this tag to scroll parent scroll bar to make that web element visible on screen.

- **locatorExpression:(mandatory)**

This attribute contains expression specific to locator type

- **locatorType:(mandatory)**

There are 4 types of locator: id, cssselector, name, and xpath which defines locator type.

- **countForVertical:(optional)**

if isdynamic is true , then this attribute can be applicable. default value is 21 (number of time to enter ARROW key). For Vertical scroll only.

- **countForHorizontal:(optional)**

if the scroll is dynamic , then this attribute can be applicable. default value is 3 (number of time to enter ARROW key) For horizontal scroll only.

- **scrollArea:(optional)**

This attribute is the locatorExpression of that scroll area where Height & Width is mention in px.This attribute is used for dynamic scroll not for static scroll.

- **direction:(optional)**

if the scroll is dynamic , then this attribute is mandatory , possible values are up, down, right,left.

- **click:(optional)**

if the scroll is dynamic this click is required, so its required to select the first item , for this this attribute is used and possible value's are right and left.Default value is left.

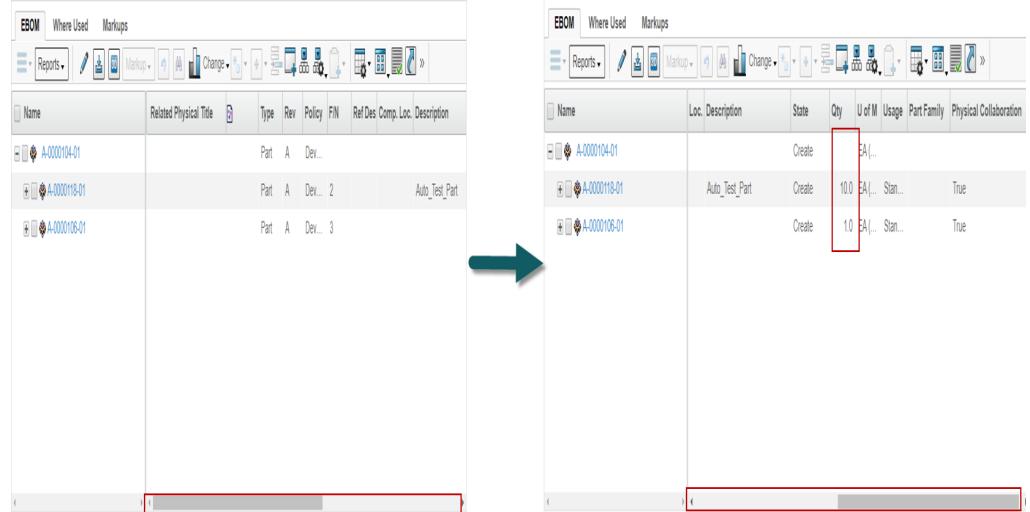
- **wait:(optional)**

This attribute contains wait time in milliseconds to find element. Default value will be taken from '3dx-tas.execution.step.timeout'. Preferred wait is 5000MS

- **Example:**

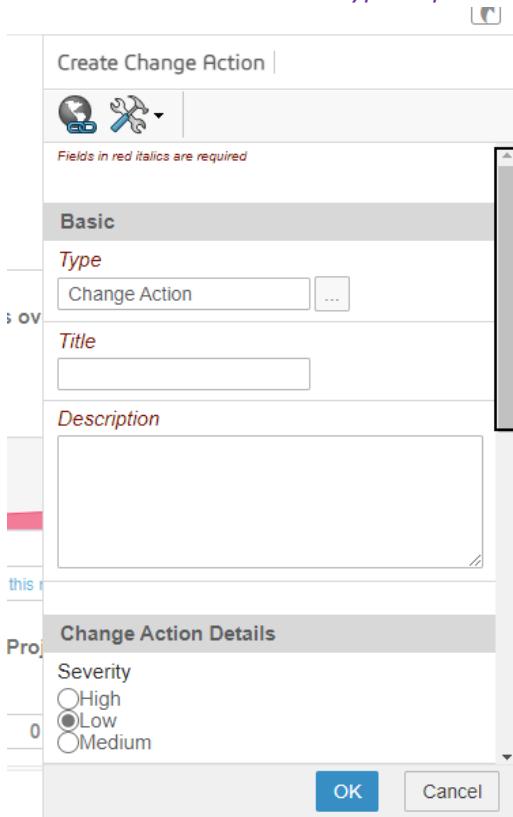
Please check below image. So, to edit quantity of row, we first must move scroll bar to right. For indented table give xpath of required of first row.

```
<ScrollToElement locatorType="xpath"
locatorExpression="//table[@id='bodyTable']//tr[@id='0']/td[@position=12]"/>
```



Please check below image. It can be used both the way to scroll horizontally and to scroll vertically to find any element.

```
<ScrollToElement locatorType="xpath" locatorExpression="//label[@for='Follower']"/>
```



And Approvals Search Standard_Space

Changed Docs My Recent Docs >

DevTools - www.steepgraph2021x.com/3dspace/common/emxNavigator.jsp?collabSpace... Elements Console Sources Network > + 10 99+

```
<tr>...</tr>
<tr id="calc_Follower"></tr>
<tr>
  <td valign="middle" width="150" class="createLabel">
    <label for="Follower">Informed Users</label> == $0
  </td>
</tr>...</tr>
```

...ime html body.slide-in-panel div#divPageBody form table.form tbody tr td.createLabel label ...

//label[@for='Follower'] 1 of 1 ^ v Cancel

Styles Computed Layout Event Listeners DOM Breakpoints Properties Accessibility >

Filter :hover .cls +

```
element.style {
```

```
td.label, td.label label, td.createLabel, td.createLabel label,
td.labelRequired, td.labelRequired label, td.createLabelRequired,
td.createLabelRequired label {
  color: #2a2a2a;
  font: 15px/18px Arial,Helvetica,sans-serif;
  margin-bottom: 0;
  text-align: left;
  line-height: 18px.
```

Create Change Action | Clear Due Date

Approvers Assignees Add Assignee Remove

Approvers Add People Add Route Template Remove

Informed Users Add Informed User Remove OK Cancel

<ScrollToElement locatorType="id" locatorExpression="Description"/>

Instance Description Design Name Revision Not State F/N Qty Unit Of Meas Description

		A		In work				
		A		In work	1	1.0	EA (each)	F

2 objects, 1 selected

Priority State : In work + Frozen 92 < >

er : 3DEXP01 3DEXP01 filed : 2022 Oct 7 16:24:03

Markups th#Description.mx_editable

Title	Instance Title	Instance Description	Design Name	Collaborative Policy	Ref Des	Description
Part...				EC Part		
Part...	PartObject 31...			EC Part		

2 objects, 1 selected

About 3DEXPERIENCE platform

File Memory Application Security Lighthouse Welcome +

```
<th id="Description" class="mx_editable ">
  <table>
    <tbody>
      <tr>
        <td id="Description" width class="mx_sort-column "> == $0
          <a href="javascript:sortTable(18)">Description</a>
        </td>
        <td id="Description_sort" width class></td>
```

Computed Styles Layout

```
element.style {
```

```
div emxUIStruct...ser.css #mv
```

SearchOutlookMail

SearchOutlookMail:

With the help of this tag, SearchOutlookMail tag is used to search the subject and open mail by providing valid emailid.

- **Syntax:**

```
<SearchOutlookMail searchMailId="" searchText="" />
```

NOTES:

1. This tag use to Search Outlook header name from Search box

- **Attributes:**

- **searchText:(Mandatory)**

This attribute is Mandatory. this attribute searchText is used to search in outlook.

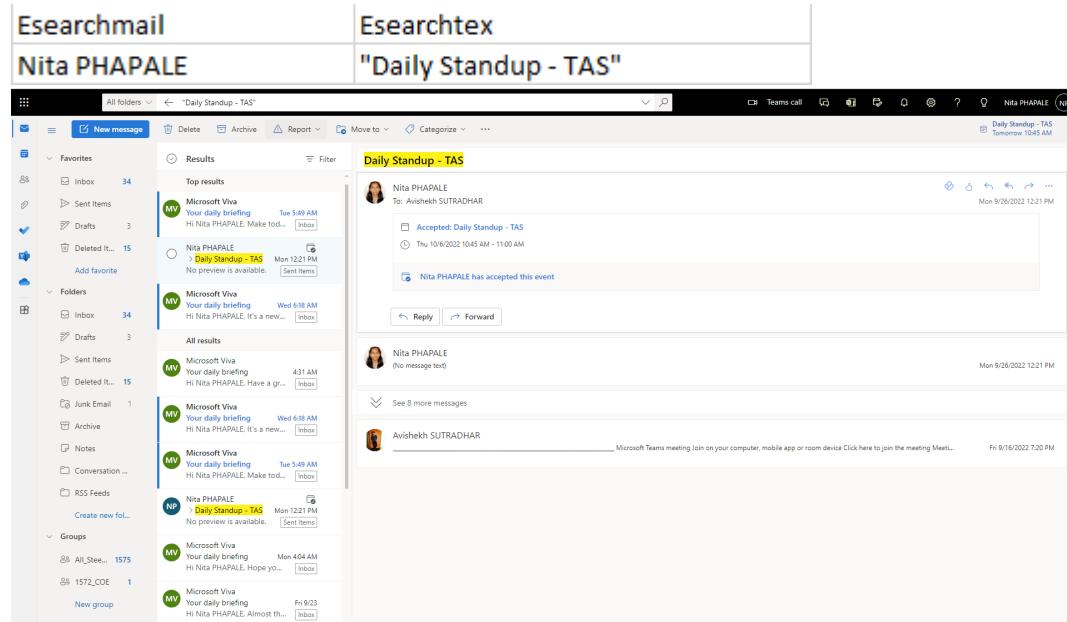
- **searchMailId:(Mandatory)**

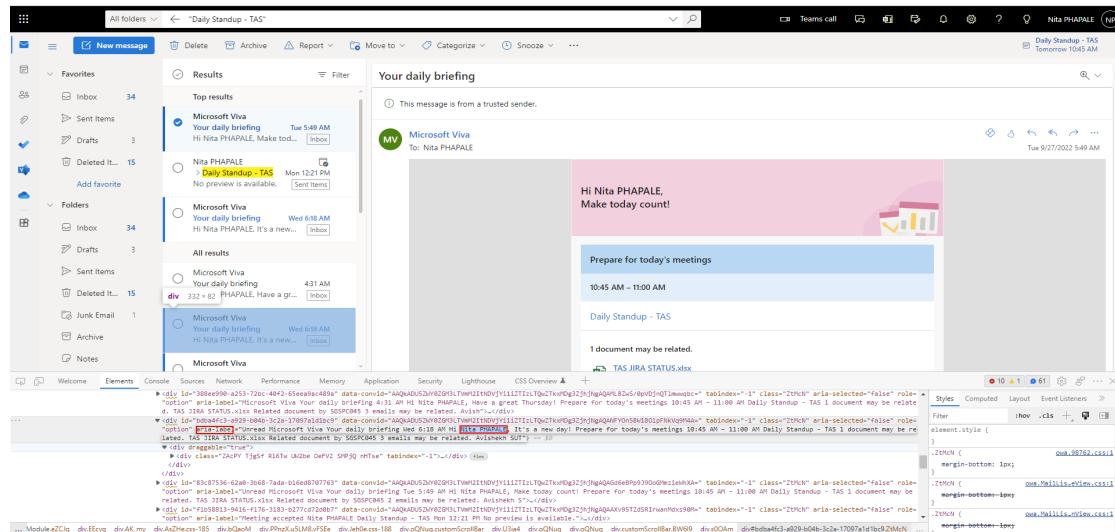
This attribute is Mandatory. this attribute searchMailId is used to search by mailId in outlook..

- **Examples:**

Suppose you want to Search mail id or Search text

```
<SearchOutlookMail searchText="$csv{Esearchtex}"  
searchMailId="$csv{Esearchmail}"/>
```





<SearchOutlookMail searchText= "Request for Material Declaration of Parts to" searchMailId="srcompleng1@ericsson.com"/>

Created with the Personal Edition of HelpNDoc: Create iPhone web-based documentation

SelectAndSubmit6WTagsSearchSelectAndSubmit6WTagsSearch

SelectAndSubmit6WTagsSearch:

This tag is for 6W tags page to select and submit the search results for further processing. This tag looks for the values provided in id attribute in the Name column and selects the checkbox for the corresponding row. If the search displays 0 results, then it clicks on clock to show results modified recently(results which were not indexed immediately).

▪ Syntax:

<SelectAndSubmit6WTagsSearch selection="" id="" submitLabel="" />

▪ Attributes:

- **selection:(mandatory)**

This attribute tells whether the items to be selected are single or multiple. It can take two values, either single or multiple.

- **id: (optional When Input)**

id attribute is used to take value to search from csv. **id** attribute is already described in [InputText](#) tag. Please check it.

- **wait(optional):**

This attribute contains wait time in milliseconds to find element.Default value will be taken from '3dx-tas.execution.step.timeout'.Preferred wait is 3000MS.

- **input(optional when id used):**

input attribute instead of id, so it can be consider as a value hard coded and it will not take from CSV values.

- **submitLabel:(mandatory)**

specifies the label of the button to submit the selected search results. User will have to specify the text seen on the submit button.

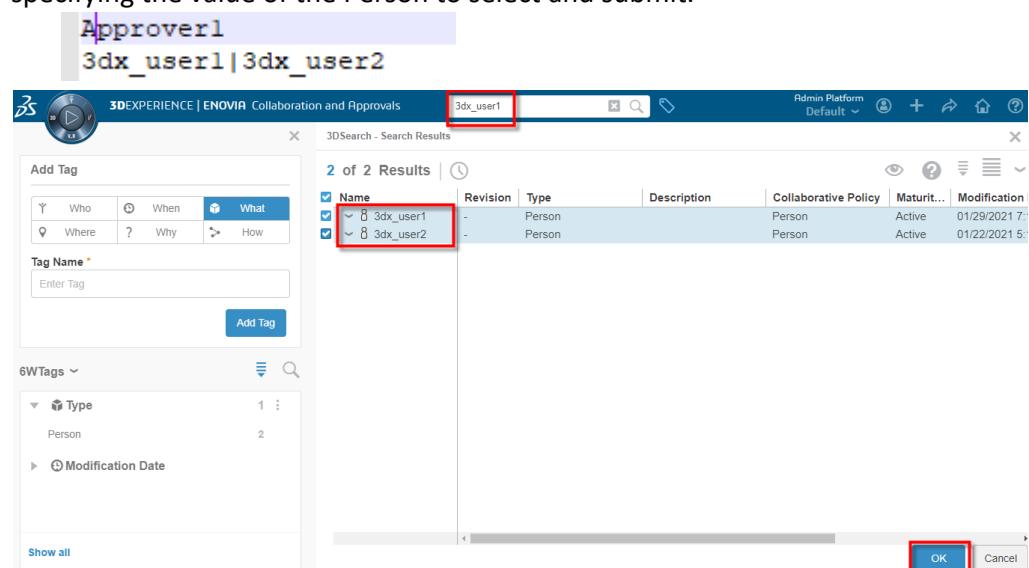
- **columnName: (optional)**
name of column header to identify the object in addition to the name.
- **columnValue:(optional)**
name of column value to identify the object in addition to the name.

▪ **Examples:**

In the image below, script has searched for a Part and now is to submit it. In this case the tag will be written as:

```
<SelectAndSubmit6WTagsSearch selection="single" selection="multiple"
id="Approver1" submitLabel="OK"/>
```

In the csv file, a corresponding header with the name Approver1 must exist specifying the value of the Person to select and submit.



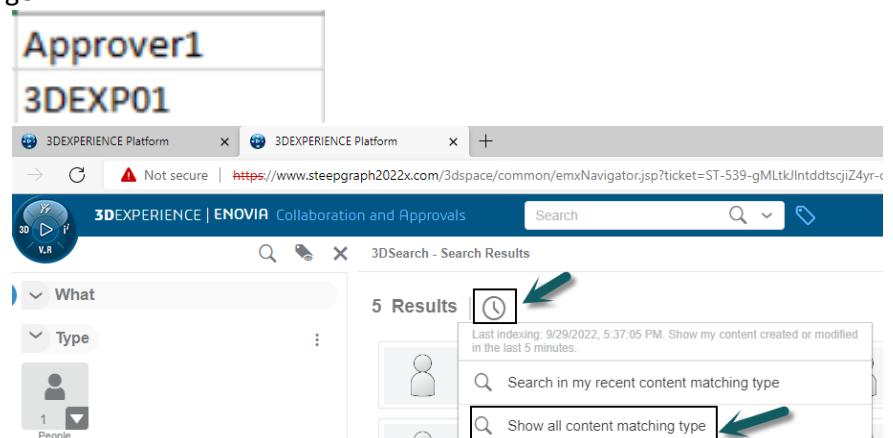
▪ **Example:**

As shown in below syntax Approver1 is Id that name you have to provide in CSV file so that it take that name from there.

```
<SelectAndSubmit6WTagsSearch selection="single" id="Approver1"
submitLabel="Ok" columnName="Type columnValue="Person"/>.
```

If some time user can not see result directly on UI than this tag Click on Clock icon, then show all content matching type

It will directly show the search result and select user for which user have search. As shown in below image.



3DSearch - Search Results									
1 of 5 Results ⏱		 ⓘ ? ⌂ ⌂ ⌂ ⌂							
Name	Title	Revision	Type	Description	Collaborat...	Maturity S...	Modificati...	Creation ...	Ov
1 ✓ 8 3DE...	3D EXP01	-	Person	Person	Active	10/14/2022...	09/18/2022...		
2 ✓ 8 admi...	Admin Plat...	-	Person	Person	Active	10/11/2022...	09/16/2022...		
3 ✓ 8 Admi...	Administrat...	-	Person	Person Obj...	Person	Active	10/03/2022...	09/16/2022...	
4 ✓ 8 Com...	Complianc...	-	Person	Person Obj...	Person	Active	09/17/2022...	09/17/2022...	
5 ✓ 8 Test ...	Test Everyt...	-	Person	Person Obj...	Person	Active	09/16/2022...	09/16/2022...	

Created with the Personal Edition of HelpNDoc: [Maximize Your Reach: Convert Your Word Document to an eBook or Kindle eBook](#)

SelectColor

SelectColor :

With the help of this tag, Select color in input (type=color) field identified using given locatorType and locatorExpression attributes.

- **Syntax:**

```
<OpenNewToolbarInDashboard toolbarName="" actionmenu="" dropdownOption="" />
```

NOTES:

- Using this tag user can select colour in input field identifier using locator expression.

- **Attributes:**

- **color:(mandatory)**

This attribute is Mandatory. Color is attribute having hexadecimal color code (e.g. #ff0000) value

- **locatorType:(mandatory)**

This attribute is Mandatory. Possible values are 'id', 'name', 'css' and 'xpath'. By using one of this Type user can give required expression

- **locatorExpression:(mandatory)**

This attribute is optional. This attribute contains expression specific to locator type

- **Examples:**

Suppose you want to write some skipped testcases.

```
<SelectColor color="#005685" locatorType="xpath" locatorExpression="//div[@class='topbar-cmd fonticon fonticon-menu menu']"/>
```

Created with the Personal Edition of HelpNDoc: [Effortlessly Edit and Export Markdown Documents](#)

SelectDate

SelectDate:

Select date from date chooser.

- **Syntax:**

```
<SelectDate locatorType="" locatorExpression="" id="" />
```

▪ **Attribute:**

- **id(optional when input used):**

id attribute is used to take value to search from csv. **id** attribute is already described in [InputText](#) tag. Please check it.

- **input(optional when id used):**

input attribute is used to provide a hard-coded value for input instead of using id attribute from csv file.

- **selecttag (optional):**

This attribute is optional. Possible values are True & false. If Select Date drop down is implemented by select tag in HTML DOM then value should be true. Default value is false.

- **locator Type(mandatory):**

There are 4 types of locator: id, CSSselector, name and XPath which are already discussed in ClickElement tag.

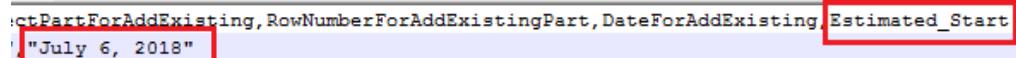
- **locatorExpression(mandatory):**

This attribute contains expression specific to locator type.

▪ **Example:**

In the image below, the date chooser next to the "Estimated Start" label has to be clicked to enter the date in the corresponding field. The tag written for this case will be:

```
<SelectDate locatorType="xpath"
    locatorExpression="//a[@id='formDateChooser']//img" id="Estimated_Start"/>
<SelectDate locatorType="xpath"
    locatorExpression="//a[@id='formDateChooser']//img" id="Estimated_Start" selecttag="true"/>
```



PartForAddExisting	RowNumberForAddExistingPart	DateForAddExisting	Estimated_Start
			"July 6, 2018"

In the csv file, a corresponding header with the name Estimated_Start must exist specifying the Date to be entered in the date chooser.

The screenshot shows a 'Create New Issue' form in a web browser. The 'Type' field is set to 'Issue'. The 'Estimated Start' field has a red border and a date picker icon. A calendar dialog is overlaid on the form, displaying the month of July 2018. The date '6' is highlighted in yellow, indicating it is selected. Other fields include 'Reported Against', 'Escalation Required' (with radio buttons for 'Yes' and 'No'), 'Priority' (set to 'Low'), and 'Problem Type' (set to 'Not Determined'). Buttons at the bottom right include 'Clear All', 'Create Issue', 'Done', and 'Cancel'.

2. In the image below, the date chooser next to the “Estimated Start” label has to be clicked to enter the date in the corresponding field. The tag written for this case will be:

To select date using input attribute user must give hard coded value. The syntax for that is given below.

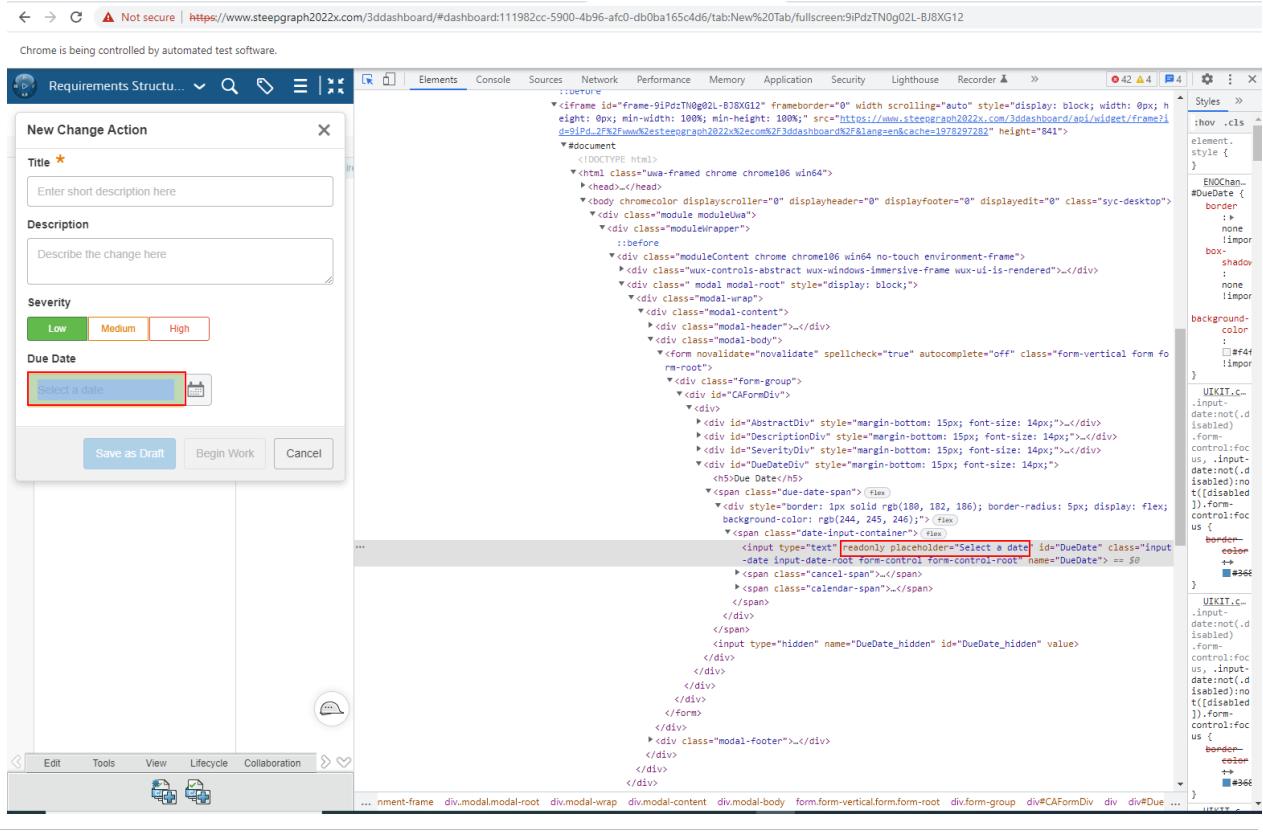
```
<SelectDate locatorType="xpath"
locatorExpression="//a[@id='formDatePicker']//img" input="Oct 7, 2022"/>
```

The screenshot shows the ENOVIA Collaboration and Approvals application. On the left, a sidebar menu includes 'Change Management', 'My Changes', 'Collaboration and Approvals' (selected), 'Dashboard', 'Issues Summary', 'Collections', 'Utilities', 'Background Jobs', 'About...', 'Routes', 'Routes Summary', 'Route Templates', 'Tasks', 'Subscriptions', 'My Calendar', and 'Meetings'. The main area displays a 'Create new Project' form for 'Standard_Space'. Fields include 'Project Space' (selected), 'AutoName' (checked), 'Select Date' (set to 'Oct 7, 2022'), 'Project Space' (dropdown), 'Currency' (set to 'USD-US Dollar-UNITED STATES OF AMERICA (THE)'), 'Description' (text area), and 'Project Scheduling' (dropdown). Below the form, a DevTools window is open over the page. The Elements tab shows an iframe containing HTML code. A specific  element is selected, highlighted with a red border. The SelectorHub tab shows several XPath and CSS selector options, with one entry for 'Rel XPath' highlighted with a red border: //img[@alt='Date Picker'].

3. For Dashboard : To select date in 3D Dashboard inspect date box in highlighted below given below to give xpath.

```
<SelectDate locatorType="xpath" locatorExpression="//input[@class='inputdate input-date-root form-control form-control-root']"
           id="Target_Release_Date" selecttag="true"/>
```

for is dashboard default value is false. After inspecting for box if it showing "select or option" as shown in picture then take `selecttag="true"`, Otherwise keep it as false .



Created with the Personal Edition of HelpNDoc: What is a Help Authoring tool?

SelectContextMenuOption

SelectContextMenuOption :

With the help of this tag, user Select option from context menu in 3ddashboard.

- **Syntax:**

```
<SelectContextMenuOption refid="" menuOption="" />
```

NOTES:

1. Using this tag user can select option from context menu in 3ddashboard.

- **Attributes:**

- **refid:(mandatory)**
id of FindElement tag will used as refid for this tag
- **menuOption:(optional)**
option number in menu. Default value is '2'

- **Examples:**

Suppose you want to write some skipped testcases.

```
<SelectContextMenuOption refid="find" menuOption="2" />
```

Created with the Personal Edition of HelpNDoc: Transform Your Help Documentation Process with a Help Authoring Tool

SelectElement

SelectElement:

Select element in the drop down list, present on the web page.use either id or input attribute.

- **Syntax:**

```
<SelectElement locatorType="" locatorExpression="" id="" />
```

- **Attributes:**

- **id :(optional when input used):**

id(csv) attribute is csv column identifier which is already discussed in InputText tag.

- **locatorExpression:(mandatory)**

This attribute contains expression specific to locator type.

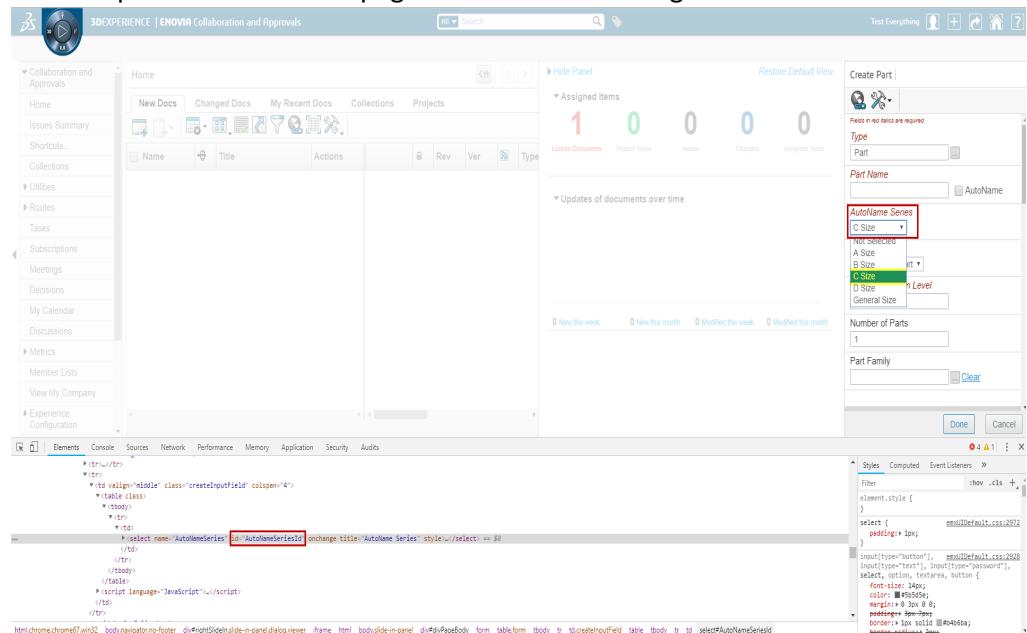
- **locatorType:(mandatory)**

There are 4 types of locator: id, cssselector, name, and xpath which defines locator type.

- **input:(optional when id used)**

This attribute is used to provide a hard-coded value for input instead of using id attribute from csv file.

Let's take an example. Suppose we would like to select Auto Name Series **C Size** from drop down on the web page shown in below image.



We should use web browser inspector tool to check attributes of web element which we want to locate. The AutoName Series dropdown has an id attribute so, we can locate this element using id.Now we must select one of the items in the dropdown. User need to pass dropdown item text in test data csv file of the test case. For that define some unique key in id attribute of this tag and use same key as csv column header. Define user input in the next row of that csv file. Here user

input is C Size.

```
<SelectElement locatorType="id" locatorExpression="AutoNameSeriesId"
id="AutoNameSeries"/>
```

CSV file looks like below:

```
"AutoNameSeries","Policy","Number of Parts",""
"C Size","EC Part","1"
```

Created with the Personal Edition of HelpNDoc: [Free EPub producer](#)

SelectIndentedTableRow

SelectIndentedTableRow:

Select indented table row to perform specific action on column of the that row.

- **Syntax:**

```
<SelectIndentedTableRow expand="" id="" criteria="" selectcheckbox=""
section="" />
<SelectIndentedTableRow id="DisplayName" criteria="text" selectcheckbox="true"
section="freezeapan" dialog="true" />
```

The action to perform is not specified by this tag. This tag only finds the indented table row with given criteria. Then subsequent tags will use reference of this tag to perform some action.

- **Attributes:**

- **wait(optional):**

This attribute contains wait time in milliseconds to find element. Default value will be taken from '3dx-tas.execution.step.timeout'. Preferred wait is 2000MS. Only applicable for 22xfd06.

Note: when criteria is hyperlink text, then wait attribute will work , or else it will take value from '3dx-tas.execution.step.timeout'.

- **selectcheckbox(optional):**

possible values are true or false. If its value is true then script will select checkbox else it will not.

Default value is true.

The screenshot shows the 3DEXPERIENCE | ENOVIA Product Specification interface. At the top, there's a toolbar with various icons. Below the toolbar, the part number A-0000104-01 is displayed along with its status (Create), owner (Test Everything), and last modified date (5/28/2018 6:31:18 PM). The main area features a navigation bar with links like 'A-0000104-01 A', 'Bill Of Materials', 'Equivalents', 'Access', 'Change Management', 'Specs & Documents', 'Related Parts', and 'Collaboration'. Below the navigation bar is the EBOM table. The table has columns for Name, Related Physical Title, Type, Rev, and Policy. There are three rows in the table:

Name	Related Physical Title	Type	Rev	Policy
A-0000104-01		Part	A	Develop...
A-0000118-01		Part	A	Develop...
A-0000106-01		Part	A	Develop...

- **expand(optional):** possible values are true or false. Default value is true. If its value is true then while traversing the levels of indented table, it will click on (+) button else it will not click. This (+) button is used to expand the current level, if the level is already expanded then it will ignore expand operation for that level.

Name	Related Physical Title	Type	Rev	Policy	F/N
A-0000104-01		Part	A	Develop...	1
+ A-0000118-01		Part	A	Develop...	2
+ A-0000106-01		Part	A	Develop...	3

- **criteria(mandatory):**

possible values are level or text Indented table row is selected based on given level or text of first column.

if this attribute is empty or null then it will throws exception.

1. **Level:**

It is starting from zero. So, level for root is 0. Comma (,) is used as separator between levels.

Suppose user want to select first child of root then value of level should be “0,0”

Similarly, if user want to select second child of root node then value of level should be “0,1”.

If level is not defined through id, then by default all the rows in the table will be selected.

Example:

Suppose user would like to select multiple levels of structure given in below image

If level = “0,0” then

EBOM Where Used Markups

Reports 3D Markup Change

Name	Related Physical Title
A-0000104-01	
A-0000118-01	
A-0000108-01	
B-0000100-01	
B-0000101-01	
A-0000106-01	
A-0000109-01	
A-0000110-01	
C-0000100-01	

EBOM Where Used Markups

Reports 3D Markup Change

Name	Related Physical Title
A-0000104-01	
A-0000118-01	
A-0000108-01	
B-0000100-01	
B-0000101-01	
A-0000106-01	
A-0000109-01	
A-0000110-01	
C-0000100-01	

If level = "0,1" then

EBOM		Where Used	Markups
	Name	Related Physical Title	
-	A-0000104-01		
-	- A-0000118-01		
-	- - A-0000108-01		
-	- - - B-0000100-01		
-	- - - B-0000101-01		
-	- - A-0000106-01	A-0000106-01	
-	- - A-0000109-01		
-	- - A-0000110-01		
-	- C-0000100-01		

If level="0,1,0" then

EBOM		Where Used	Markups
	Name	Related Physical Title	
-	A-0000104-01		
-	- A-0000118-01		
-	- - A-0000108-01		
-	- - - B-0000100-01		
-	- - - B-0000101-01		
-	- A-0000106-01		
-	- - A-0000109-01	A-0000109-01	
-	- - A-0000110-01		
-	- C-0000100-01		

The value of level for each row shown in below image.

Name	Related Physical Title
A-0000104-01	0
A-0000118-01	0,0
A-0000108-01	0,0,0
B-0000100-01	0,0,0,0
B-0000101-01	0,0,0,1
A-0000106-01	0,1
A-0000109-01	0,1,0
A-0000110-01	0,1,0,0
C-0000100-01	0,1,0,1



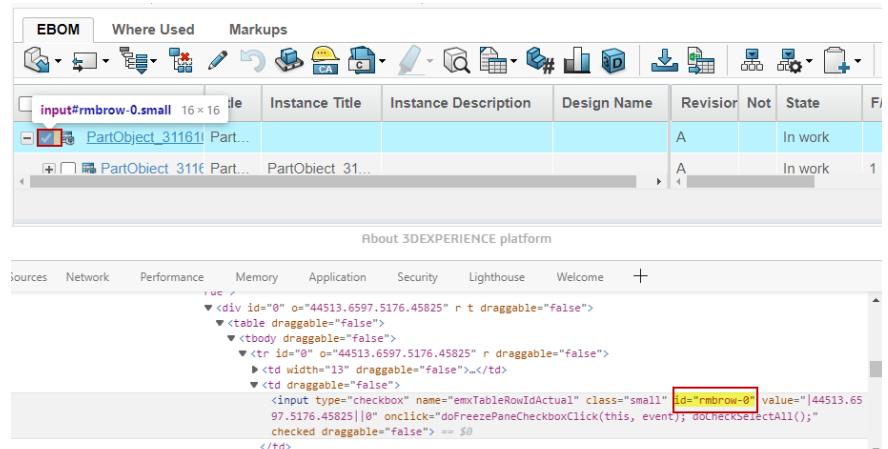
Level

- **Example:** User can select row with level attribute. User Have to pass id value in csv file. As shown below.

```
<SelectIndentedTableRow expand="false" id="IT1" criteria="level"
isNew="true" />
```

As shown in above image how we can find level of that row. That level value of row should be pass in csv file. As shown in above content.

As shown in the below image we inspect that row and find level of that row.



The screenshot shows a table with several rows. The first row has a checkbox in the first column. The second row has a checkbox in the first column and is highlighted with a red border. The third row has a checkbox in the first column. Below the table is a developer tools screenshot showing the DOM structure of the table row, specifically focusing on the first column's td element which contains an input checkbox with id='rmrow-0'.

2. text:

Select indented table row based on text of first column of the row.

Please check below image

If text="A-0000118-01" then

Name	Related Physical Title
A-0000104-01	
A-0000118-01	
A-0000106-01	

expand attribute will not be used if text criteria are used. User must expand to given level using [ClickElement](#) tag.

3.hyperlinktext :

This criteria is used when the searched text is hyperlink. for this , text assertion will happen only on the first col of non freeze pane.

Note:When criteria is hyperlinktext,then wait attribute will work ,or else it will take value from '3dx-tas.execution.step.timeout'.

Instan	Title	Qty	Rev	Type	State
-	Manufacturing Assembly00000018	1	A	Manufact...	Fro...
+	Manufacturing Assembly00000019	1	A	Manufact...	In ...
+	Transform00000011	1	A	Transform	In ...

- id(mandatory when input not used):** It is csv column identifier. So, value of level or text should give in csv file.
- input(mandatory when id not used):** This is optional attribute if Id attribute is used. This attribute is used to provided the value of level or text hardcoded in tag.
- section (optional):**
If criteria is "text" then only section tag will be of use. Possible values for section tag are "table" and "freezeapan".
Default value is table.
 - table :** select indented table row with given text from the right of the freezeapan. If text="Testlab-04" then

Test Labs

ID	Name
<input checked="" type="checkbox"/> TL-04	Testlab - 04
<input type="checkbox"/> TL-03	Testlab - 03

2. **freezepan:**

select indented table row with given text from left of the freezepan. If no section tag defined and criteria is text then default section tag value is table.

Name	Related Physical Title
-	A-0000104-01
+	<input checked="" type="checkbox"/> A-0000118-01
+	A-0000106-01

- **Example:** User can select intended table row using text attribute as shown below

```
<SelectIndentedTableRow expand="false" id="object" criteria="text"
```

```
selectcheckbox="true" section="freezepan" dialog="false"/>
```

Name	Description	State	Scope	Owner
<input checked="" type="checkbox"/> ML4	Member List of Assembly ...	Inactive	Personal	3DEXP01 3DEXP01
<input type="checkbox"/> ML5	Member List of Assembly ...	Inactive	Personal	3DEXP01 3DEXP01
<input type="checkbox"/> ML-000102		Active	Personal	3DEXP01 3DEXP01
<input type="checkbox"/> ML9	Member List of Assembly ...	Active	Personal	3DEXP01 3DEXP01

To Select this row we use selectIndentedTableRow tag. To select we have to give some attribute value in csv file. we have to

provide id and its value in csv file as shown below.

object ML 4

Select table row with text attribute we have to add section attribute in it. As shown above in tag.

- **dialog(optional):**

This attribute is for table in dialog window. For indented table in dialog window, set this value to 'true'. Possible values are 'true' or 'false'. Default value is 'false'.

- **linkValue(optional):**

The possible values are 'true' or 'false'. Default value is 'false'. This attribute is for case where there is link in the column value.

- **isnew(optional):**

The possible values are 'true' or 'false'. Default value is 'false'.

When new row is added in table and we have to select that row at that time we will keep value is

true and if not that we will keep its value as false. As shown in above

example.

Note:

If user is creating table structure and traversing in it one after other, in such cases refresh indented table every time new child is added to structure and before selecting row because 3DExperience will assign row number properly after refresh.

Created with the Personal Edition of HelpNDoc: [Full-featured Kindle eBooks generator](#)

SelectorOpenNewIndentedTableRow

SelectorOpenNewIndentedTableRow:

User use this tag to Open or Select newly created indented table row **after creation**.

Syntax:

`<selectoropennewindentedtablerow criteria="" />`

Note:

This Tag is applicable only for 3DSpace

Attribute:

- **criteria(mandatory):**

possible values are open or select.

select is used to select the object and Open is used to open the object.

- **searchByIdentifier(optional):**

possible values are 'true' or 'false'. default value is false.

by using this attribute , it can open or select the new object in Indented table.



using this Search it will select or open the object.

- **id(optional):**

used only if searchByIdentifier is 'true'. id(CSV) attribute is mapped to csv file column header. Input value should be taken from csv file which has to be defined by user in csv. This value will be used to search for the object which user wants to select or open.

The screenshot shows a software interface with two main sections. The top section is a table with columns: Name, Title, Instance Title, Instance Description, Design Na, Revision, Not, State, F/N, Qty, Unit Of Meas, Des. There are three rows: the first row has an unchecked checkbox and the text 'PartObject_31161 PartObject...'; the second row has a checked checkbox and the text 'PartObject_31161 PartObject...'; the third row has a checked checkbox and the text 'Part.1'. The bottom section is a detailed view of a part object named 'PartObject_31161063_000061...'. It shows tabs for Properties, Images, Revisions, Customer Extension, and Relate. Under the Properties tab, there are sections for Basics, Classification Path(s), and Structure Content. The Basics section includes fields for Type (Part), Name (PartObject_31161063_0000618_01), Title (Part), Description (Part for Machinery), Revision (A), State (In work), and Creation (Oct 20, 2022). The Classification Path(s) section lists 'PartObject_31161063_000061... A Engineering BOM Materials & Composition Equivalents Change Management Part Specs & Documents Related Parts Lifecycle and Collaboration Characteristics Tables Classification PowerView IP and Export Control'.

Example :

```
<selectoropennewindentedtablerow criteria="Open"/>
<SelectOrOpenNewIndentedTableRow criteria="open" id="searchObject"
searchByIdentifier="true"/>
```

Created with the Personal Edition of HelpNDoc: [Streamline Your CHM Help File Creation with HelpNDoc](#)

SelectSecurityContext

SelectSecurityContext:

SetSecurityContext tag let's you change the security context from My credentials (i.e.Organization, Collaborative Space and Role) . It has three attributes: organization, collaborativeSpace and role.

- **Syntax:**

```
<SelectSecurityContext organization="" collaborativeSpace="" role="" />
```

Attributes:

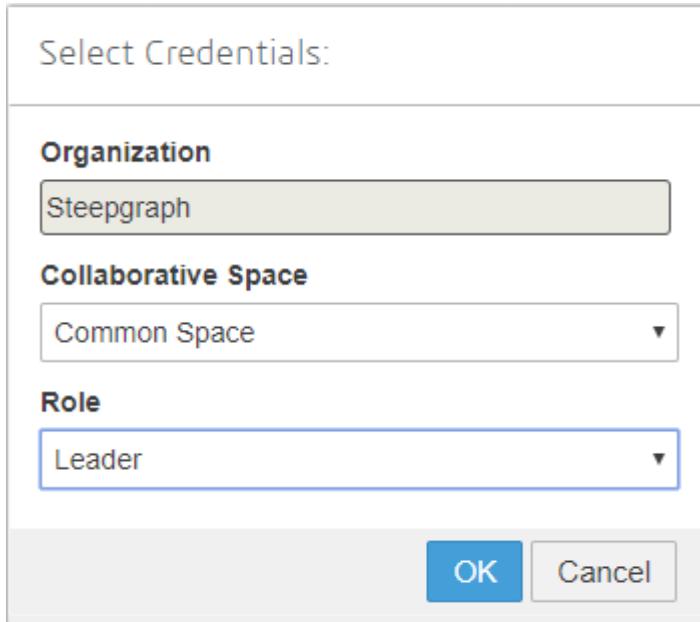
- **organization:**
(mandatory): Used to provide organization value for Security context.
- **collaboarativeSpace:**
(mandatory): Used to provide Collaborative space value for Security context.
- **role:**
(optional): Used to provide role value for Security context.

Example:

Suppose the user wants to set value of Organization, Collaborative Space and Role to Steeograph, Common Space and Leader respectively.

The tag in this case will be as follows:

```
<SelectSecurityContext organization="Steeograph" collaborativeSpace="Common
Space" role="Leader"/>
```



Created with the Personal Edition of HelpNDoc: [Produce Kindle eBooks easily](#)

SelectWindowRegion

SelectWindowRegion:

User can use selectwindowregion tag when he wants to get any element from screen with specific region.

All attributes are mandatory.

- **Syntax:**

```
<SelectWindowRegion imagepath="" skiperror="" id="" />
```

Attributes:

- **imagepath:(Mandatory)**

Specifies the path where the image of the screen element is located. This is mandatory attribute.

- **skiperror:(Optional)**

Specifies whether to skip current test case or not. Values are true or false. Should be set to false to run test case. Default value is false.

- **id: (Mandatory)**

This attribute is used to take value to search from csv. id attribute is already described in InputText tag. Please check it. This is mandatory attribute.

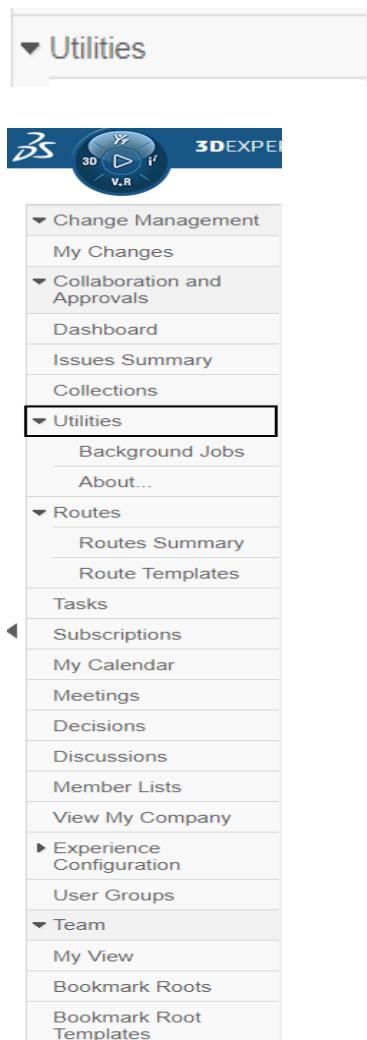
- **wait:(optional)**

This attribute contains wait time in MS to find element. default wait will be taken from 3DX_TAS propertie's "3dx-tas.execution.step.timeout" key. Preferred wait is 10000.

Examples:

If user wants to click Utilities button on UI then user needs to give an image of "Utilities" button in which it resides .

```
<SelectWindowRegion imagepath="C:\Users\SGSPC044\Pictures\Utilities.PNG"
skiperror="true" id="Test"/>
```



■ **Example:**

If screen contains two “Ok” buttons in that case user needs to give image of region in which “Ok” button resides.

```
<SelectWindowRegion imagepath="C:\Users\SGS-09\Desktop\Test.png"
skiperror="true" id="Test"/>
```

Created with the Personal Edition of HelpNDoc: [Free iPhone documentation generator](#)

SendKeys

SendKeys:

With help of this tag, user could enter text in form of keystrokes sent to an element in an application using given key attribute.

For example: Suppose user wants to enter some text in Password field. Firstly user clicks on Password field and would enter text(i.e. user could send keystrokes(i.e.required text) to the an element(i.e. Password field)).

NOTE: if the key is not valid then it will pass as text to the web element.

■ **Syntax:**

```
<SendKeys key="" />
```

■ **Attributes:**

- **key: (mandatory)**

This attribute refers to text which needs to be entered by user for an element, in form of keystrokes

Possible values for key are :

"CANCEL", "HELP", "BACK_SPACE", "TAB", "CLEAR", "RETURN", "ENTER", "SHIFT", "LEFT_SHIFT", "CONTROL", "LEFT_CONTROL", "ALT", "LEFT_ALT", "PAUSE", "ESCAPE", "SPACE", "PAGE_UP", "PAGE_DOWN", "END", "HOME", "LEFT", "ARROW_LEFT", "UP", "ARROW_UP", "RIGHT", "ARROW_RIGHT", "DOWN", "ARROW_DOWN", "INSERT", "DELETE", "SEMICOLON", "EQUALS", "NUMPAD0", "NUMPAD1", "NUMPAD2", "NUMPAD3", "NUMPAD4", "NUMPAD5", "NUMPAD6", "NUMPAD7", "NUMPAD8", "NUMPAD9", "MULTIPLY", "ADD", "SEPARATOR", "SUBTRACT", "DECIMAL", "DIVIDE", "F1", "F2", "F3", "F4", "F5", "F6", "F7", "F8", "F9", "F10", "F11", "F12", "A

part from above mentioned key you can pass any key with combination of above.

For example: SHIFT+"AnytextValue" CONTROL+a

NOTE: if the key is not valid then it will pass as text to the web element.

- **id :(Optional)**

id attribute is csv column identifier which is already discussed in InputText tag.

- **locatorExpression:(mandatory)**

This attribute contains expression specific to locator type

- **locatorType:(mandatory)**

There are 4 types of locator: id, cssselector, name, and xpath which defines locator type.

- **Example:**



ENTER_PASSWORD

Syntax:

```
<ClickElement locatorImage="125569878788.png"/>
<SendKeys key="Pass1"/>
```

Note: send key tag is already explained in web-locate element and perform action-tag no 11.

Created with the Personal Edition of HelpNDoc: [Benefits of a Help Authoring Tool](#)

Setstartwindow

Setstartwindow:

User can use Setstartwindow tag when there are multiple windows opened and he want to set one of them as start window.

- **Syntax:**

```
<SetStartWindow/>
```

- **Example:**

As shown in below image if you have to come in starting window you have to first set your starting window.

<SetStartWindow/>

Created with the Personal Edition of HelpNDoc: [Make Documentation Review a Breeze with HelpNDoc's Advanced Project Analyzer](#)

ShowNotification

ShowNotification:

Show custom notification message in log file.

- **Syntax:**

<ShowNotification message="" />

- **Attributes:**

- **message: (mandatory)**

this attribute is used to pass the message which will be printed in the log file.

Examples:

The user wants to show custom notification message post processing a tag. The tag in this case will be as follows:

<ShowNotification message="Tag processed" />

This message will be printed in the log file.

Created with the Personal Edition of HelpNDoc: [Effortlessly Publish Your Word Document as an eBook](#)

StartPerformanceLogs & StopPerformanceLogs

StartPerformanceLogs & StopPerformanceLogs:

These tags are used to get the performance in the form of time during execution of script. These tags capture the requests in between these tags and give the performance time for a script or a part of script.

- **Syntax:**

<StartPerformanceLogs />

<StopPerformanceLogs startRequest="" stopRequest="" requestOccurrence="" />

NOTE:

1. All the attributes are optional in StopPerformanceLogs tag.
2. Time is calculated from time difference between start time of requests of startRequest and stopRequest attributes. Performance time and logs are also recorded in a separate Performance log file.
3. If startRequest and stopRequest attributes are not passed, then performance time is calculated as time of all the requests in between both tags.

■ **Attribute:**

- **startRequest:(optional)**
used to provide first occurrence of a request containing passed value. Time is calculated from this point.
- **stopRequest:(optional)**
used to provide last occurrence of the request containing passed value. Time is calculated upto this point.
- **requestOccurrence:(optional)**
By giving this value, we can decide the occurrence of the stoprequest in the recorded logs. Possible values are 'first' or 'last', default value is 'first'.

■ **Example:**

```
<StopPerformanceLogs startRequest="emxIndentedTable.jsp"
stopRequest="3dspace/common/images" requestOccurrence="last"/>
```

Created with the Personal Edition of HelpNDoc: [Transform Your CHM Help File Creation Process with HelpNDoc](#)

SwitchToContentFrame

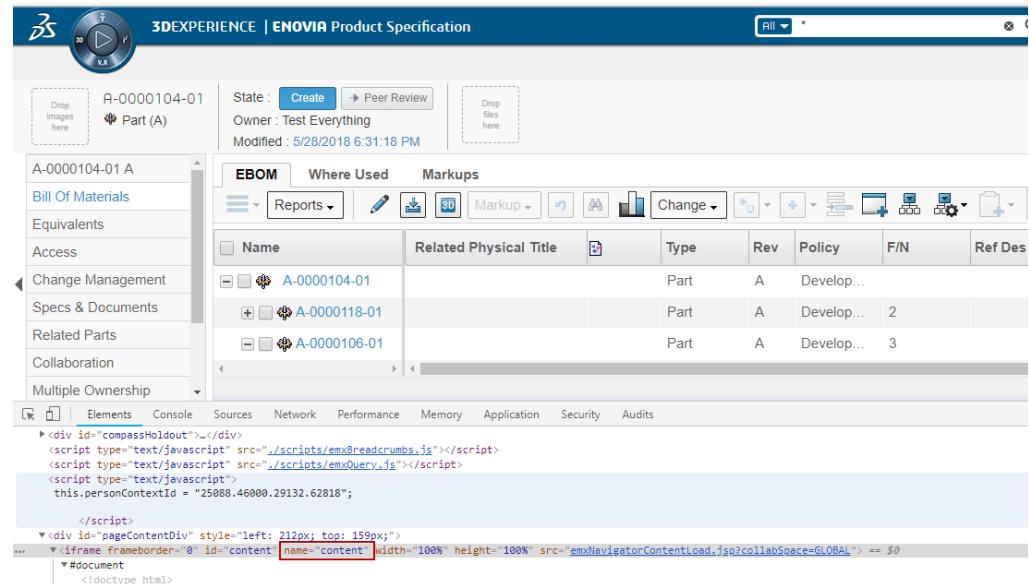
SwitchToContentFrame:

Move script control to content frame.

■ **Syntax:**

<SwitchToContentFrame/>

This is standard frame of 3DEXperience. You can also use [SwitchToFrame](#) tag to move script control that frame but in that you need to pass frame name.



Created with the Personal Edition of HelpNDoc: [Full-featured EPub generator](#)

SwitchToDefaultContent

SwitchToDefaultContent:

To return script control back to default from any frame.

■ **Syntax:**

<SwitchToDefaultContent/>

If required web element is not in any frame and current script control belongs to

some frame, then to access this element first we need to come out all the frames.

■ Examples:

Please check below image:

```
<SwitchToFrame name="ENCBOM"/>
<ClickElement locatorType="xpath" locatorExpression="//td[@id='ENCInderedBOMEeditAll']/img"/>
<SwitchToDefaultContent/>
<ClickElement locatorType="cssselector" locatorExpression="next"/>
```

You can see in above image, script control first switch to ENCBOM frame, clicks on the web element identified by given xpath in that frame. Now web element with class name as "next" is not in any frame so to access that element we first need to come out of ENCBOM frame.

Created with the Personal Edition of HelpNDoc: [Effortlessly Create High-Quality Help Documentation with a Help Authoring Tool](#)

SwitchToDetailsDisplayStyle

SwitchToDetailsDisplayStyle:

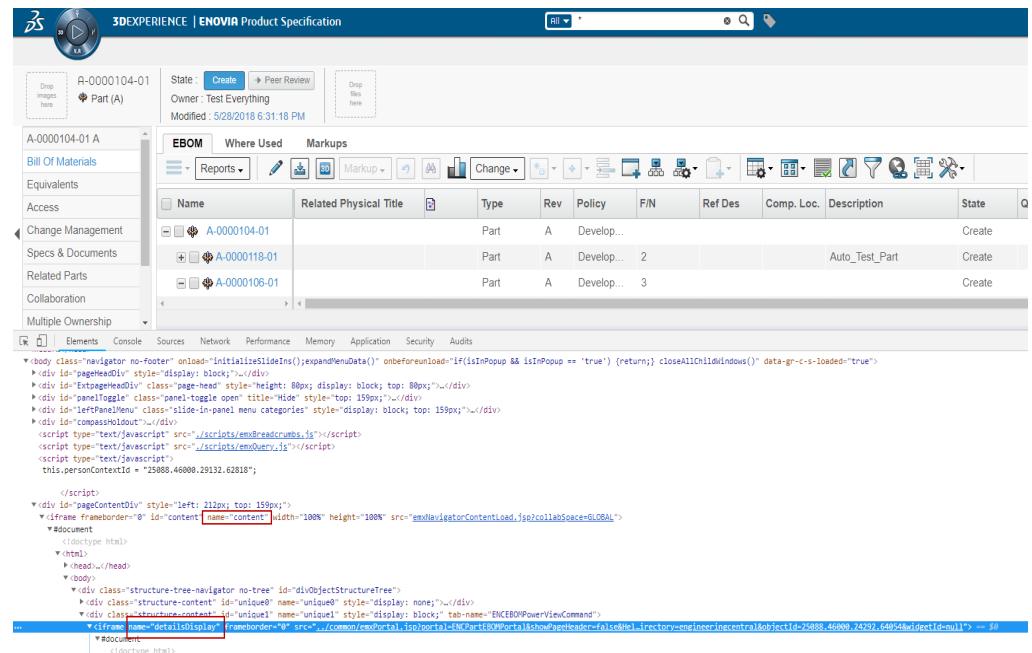
Move script control to details display frame.

■ Syntax:

```
<SwitchToDetailsDisplayStyle/>
```

This is also standard frame of 3DExperience. You can also use [SwitchToFrame](#) tag to move script control that frame but in that you need to pass frame name.

This frame is always inside the content frame. So, when you use this tag, the control will first move to content frame and then details display frame.



Created with the Personal Edition of HelpNDoc: [Upgrade your help files and your workflow with HelpNDoc's WinHelp HLP to CHM conversion](#)

SwitchToFrame

SwitchToFrame:

To access element inside given frame, script control needs to be switched to that frame.

■ **Syntax:**

`<SwitchToFrame name="" refid="" />`

Note: either name, id or index anyone should be mentioned.

■ **Attributes:**

- **name:(optional)**

This attribute defines the name of the frame. If frame name is not defined, then user can use, id of frame element as name

- **refid: (optional)**

id attribute of FindElement tag will used as refid for this tag

- **locatorType:(optional)**

there are 4 types of locator:id, cssselector, name and xpath which defines locator type.

- **locatorExpression:(optional)**

This attribute contains expression specific to locator type

- **removeSpace:(optional)**

This attribute removes extra space from the value if set to 'true'. possible values are 'true' or 'false'

- **index:(optional)**

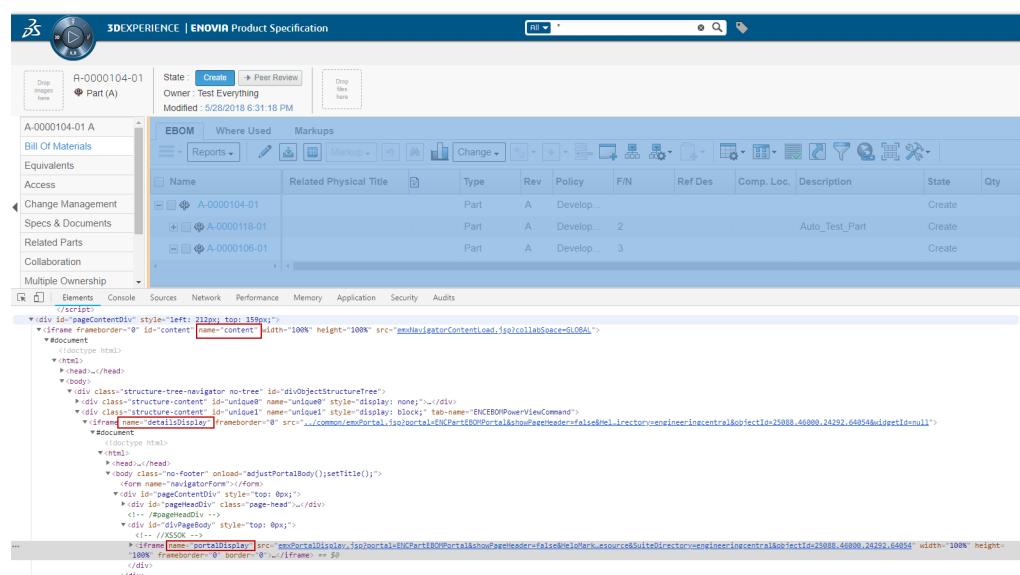
If frame to be switched does not have a name or id, it can be located relatively using index number of the frame

■ **Examples:**

parentframename=>childframename

Please check below image. To switch to portalDisplay, we need to first switch to content frame, then detailsDisplay first. So tag should be defined like

`<SwitchToFrame name="content=>detailsDisplay=>portalDisplay"/>`



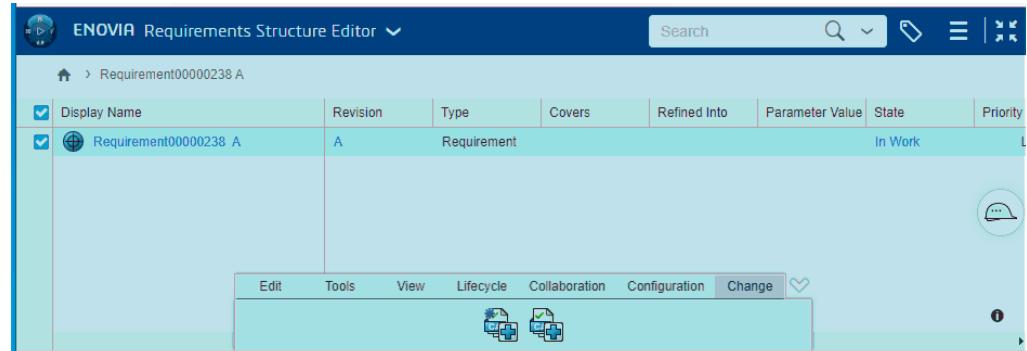
For 3DDashboard

■ **Examples: in dashboard if iframe name is not found,user need to take iframe**

name by inspecting that frame and value has to be taken from src for better understanding refer image higlighted area

- user need to add id values in csv file

```
<SwitchToFrame locatorType="xpath"
locatorExpression="//iframe[contains(@src,'ENORETV_AP')]"/>
```



Created with the Personal Edition of HelpNDoc: [Create iPhone web-based documentation](#)

SwitchToParentFrame

SwitchToParentFrame:

Returns script control back to the parent frame of current frame

- **Syntax:**

```
<SwitchToParentFrame/>
```

If the script is currently inside detailsDisplay frame, then the control will to the parent frame,i.e content frame on using the SwitchToParentFrame tag.

- **Examples:**

Please check below image:

```
<SwitchToFrame name="ENCBOM"/>
<ClickElement locatorType="id" locatorExpression="displayModeMenu"/>
<ClickElement locatorType="xpath" locatorExpression="//label[contains(text(),'Table')]" />
<SwitchToParentFrame/>
<ClickElement locatorType="xpath" locatorExpression="//div[@id='divPvChannel-1-1']//div[@class='window-size btn-maximize']"/>
```

You can see in above image, script control first switch to ENCBOM frame, clicks on the web element identified by given xpath in that frame.

Now web element with xpath : as `//div[@id='divPvChannel-1-1']//div[@class='window-size btn-maximize']` is in portalDisplay frame, which is the parent of ENCBOM frame.

Created with the Personal Edition of HelpNDoc: [Modernize your help files with HelpNDoc's WinHelp HLP to CHM conversion tool](#)

SwitchToParentWindow

- :

Move script control to parent window.

■ **Syntax:**

<SwitchToParentWindow/>

■ **Examples:**

Sometimes when we click on submit button, 3DEXperience closes current pop window. In such cases we need to switch script control back to parent window to proceed further in test case.

This can be done using this tag. User can also use [SwitchToWindow](#) tag to switch control to any other window.

Created with the Personal Edition of HelpNDoc: [Easy to use tool to create HTML Help files and Help web sites](#)

SwitchToPortalDisplayFrame

SwitchToPortalDisplayFrame:

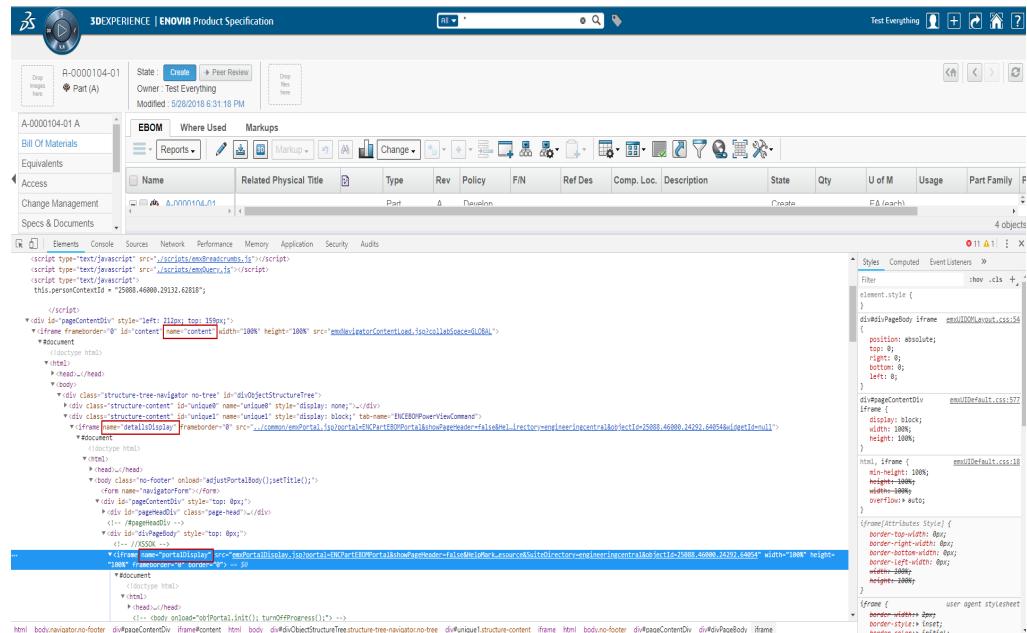
Move script control to portal display frame.

■ **Syntax:**

<SwitchToPortalDisplayFrame/>

This is also standard frame of 3DExpereience.

This frame is always inside the details display frame. So, when you use this tag, the control will first move to content frame then details display frame and then to portal display frame.



Created with the Personal Edition of HelpNDoc: [Streamline Your CHM Help File Creation with HelpNDoc](#)

SwitchToSlideInWindow

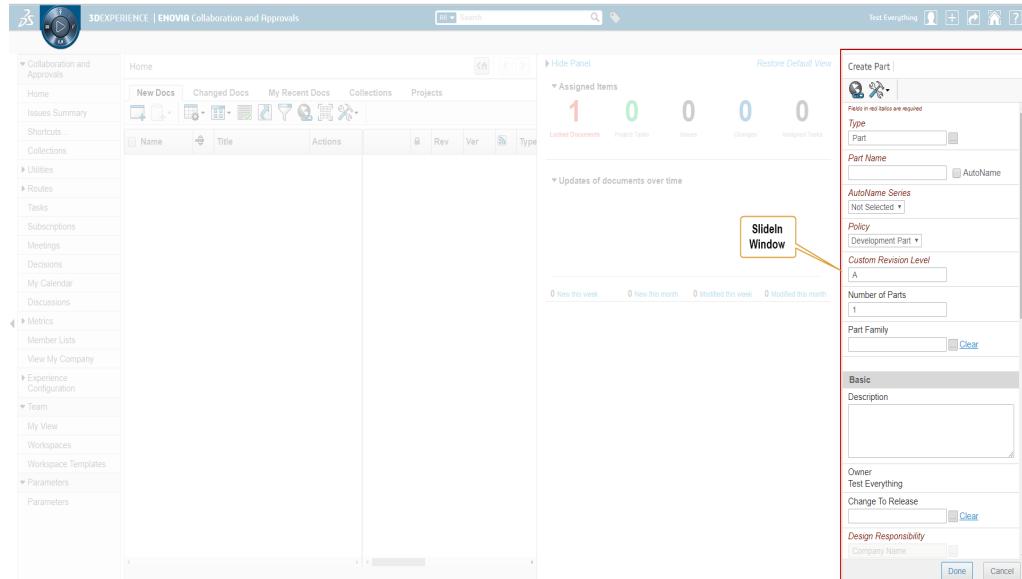
SwitchToSlideInWindow:

Move script control to slidein frame.

■ **Syntax:**

<SwitchToSlideInWindow/>

This is actually frame but generally we call it as slidein window that the reason tag is named as slideinwindow. Please check below image.



Created with the Personal Edition of HelpNDoc: [Effortlessly upgrade your WinHelp HLP help files to CHM with HelpNDoc](#)

SwitchToStartWindow

SwitchToStartWindow:

As discussed in Setstartwindow tag, window which is tagged as start window user can able to switch same window by Switch to start window tag.

This tag is only use when use have Set your Starting window in that browser.

- **Syntax:**

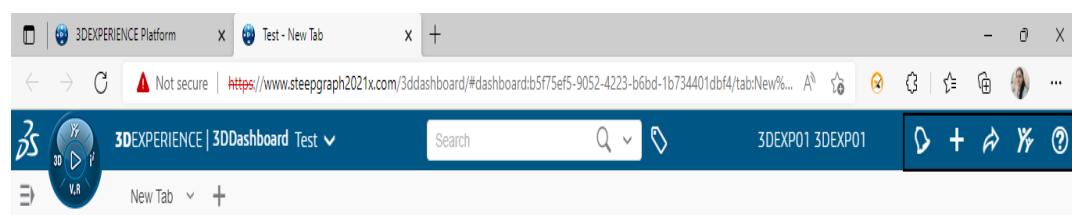
<SwitchToStartWindow/>

- **Example:**

<SwitchToStartWindow/>

As show in the below Script.If you have to logout ENOVIA Account and you are in 3DDashboard then you have to first go to starting window because there is

no other direct option for logout in 3DDashboard UI As shown in image 3DDashboard have no logout option



Created with the Personal Edition of HelpNDoc: [Experience a User-Friendly Interface with HelpNDoc's Documentation Tool](#)

SwitchToWindow

SwitchToWindow:

Pass script control to given window.

When there is more than one window then user needs to make sure, the web element he/she tries to access and script control on same window. Sometime when we click on some web element, it will open new window. To access element on this newly opened window, script

control needs to switch to that window. In such cases, we need to use this tag. If 2 tabs have same title, the driver then switches to browser tab different from current tab for this tag.

- **Syntax:**

`<SwitchToWindow title="" last="">`

- **Attributes**

- **title:(Optional)**

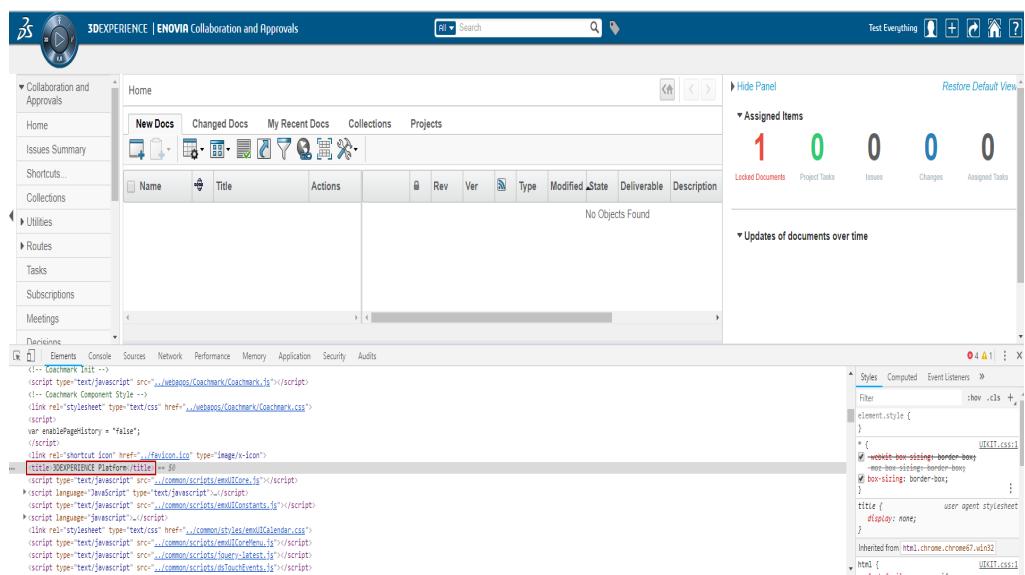
If not provided title name then it will be considered the default current window title name.but mostly give window title name means you will know the exact title name of the window.

- **last:(Optional)**

possible values are true or false.Default value is false.

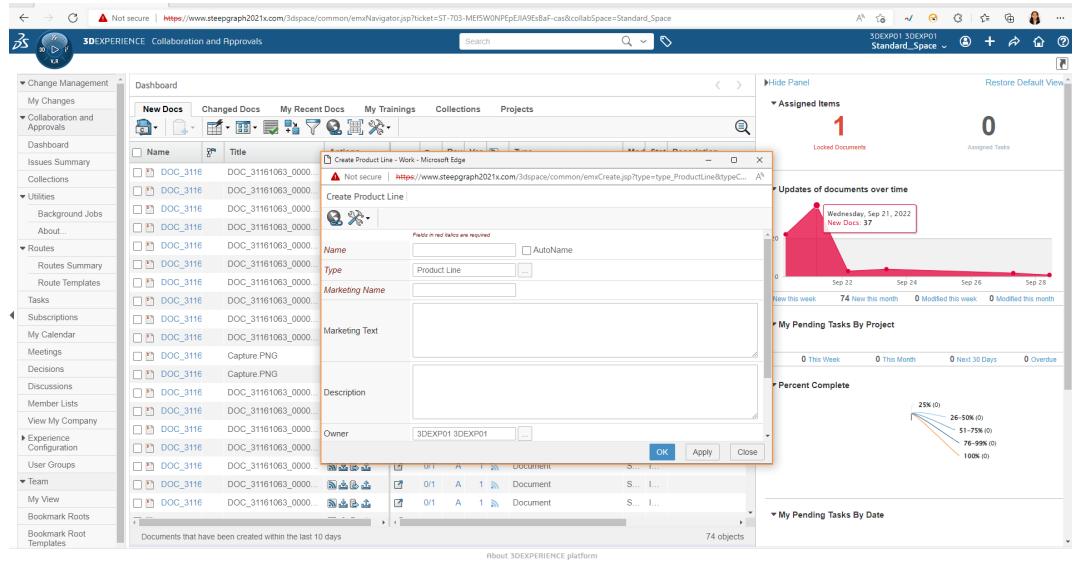
- **onLoadCompletion:(Optional)**

possible values are true or false, default value is true. true is to switch to a window where the loading is complete and false is to switch to a window where the loading is still going on.



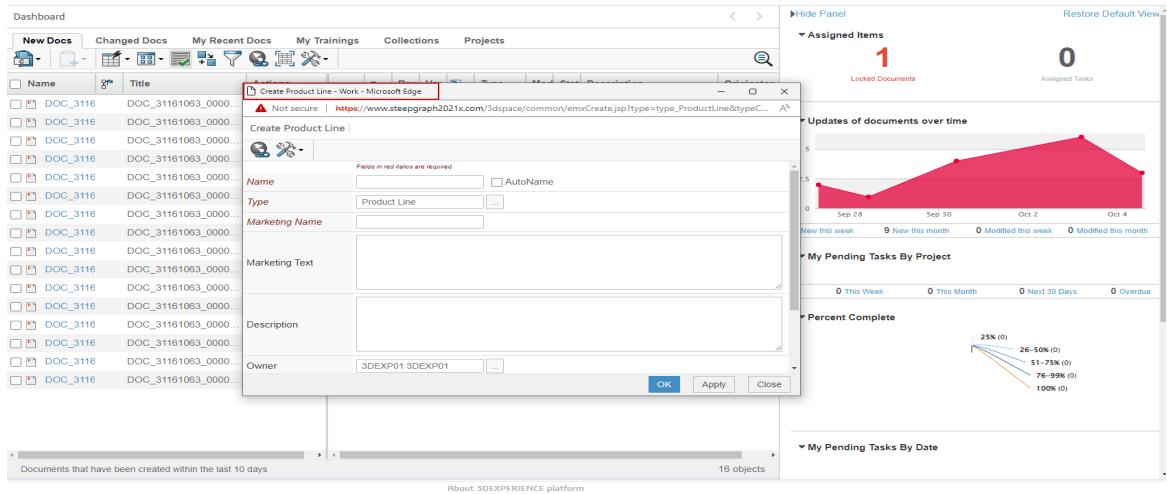
- 1) Title represent title of window to switch. You can inspect web page and find the title of page like.If title is empty, then it will switch to any other available window. If there are only two windows available, in this case you can switch between window without using title but if window count is more than two, then title of window is necessary to make sure script is switching to correct window.

When user wants to switch from one window to another another window then this can be used.



2. User can switch to new window by giving the name of that window .For e.g if user wants to switch to new window "Create Product Line " then syntax will be like this :<*SwitchToWindow title="Create Product Line" last="true"/>*>

The name of that window is highlighted in the image given below



Created with the Personal Edition of HelpNDoc: [Generate Kindle eBooks with ease](#)

SetContent

SetContent:

Set content of element identified using given, locatorType and LocatorExpression. Input value should be taken from CSV. id attribute is map to CSV column header. It work same as InputText Tag, **use either id or input any one attribute.**

- **Syntax:**

<*SetContent locatorType="id" locatorExpression="" id=""*>

- **Attributes:**

- **locatorType:(mandatory)**

This attribute is already defined for ClickElement tag. User could refer

to same.

- **locatorExpression:(mandatory)**

This attribute is already defined for ClickElement tag. User could refer to same.

- **id:(Optional when using input)**

Here id attribute is just a unique key that will referred by other tags to perform action on the web element located by this tag.

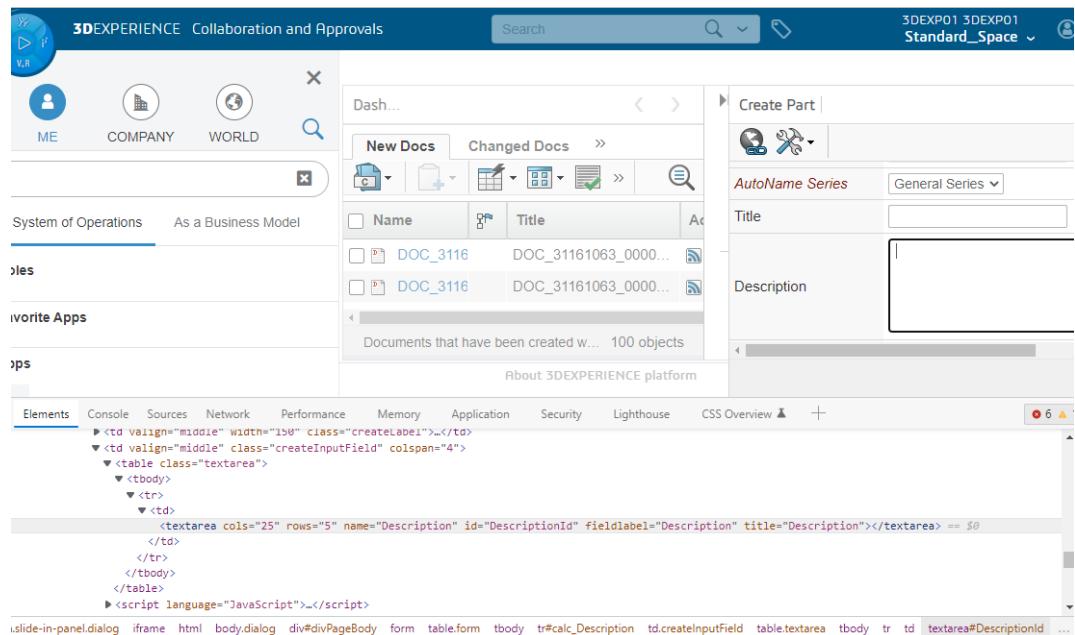
- **input:(Optional when using id)**

This attribute is used to provide a hard-coded value for input instead of using id attribute from CSV file

- **Example**

Set content In set content tag you have to provide id of particular element in which you have to give input and that input have to provide in CSV file as shown in below tag and image

```
<SetContent locatorType="id" locatorExpression="DescriptionId"
id="Content"/>
```



Created with the Personal Edition of HelpNDoc: [Make Your PDFs More Secure with Encryption and Password Protection](#)

SelectAndSubmit

SelectAndSubmit: (Note This tag is deprecated from 21x version and please refer [SelectAndSubmit6WSearch tag](#).)

User can use Submit the search results for further processing.

- **Syntax:**

```
<SelectAndSubmitSearch selection="" id="" criteria="" submitLabel="" />
```

- **Attribute:**

- **selection:(mandatory)**

This attribute tells whether the items to be selected are single or

multiple. It can take two values, either single or multiple.

- **id:(mandatory)**
(mandatory) :id(CSV) attribute is mapped to csv file column header, id attribute is used to take value to search from csv
 - **selection:(mandatory)**
There are 2 types of values:
 - **text** : User can use this text value after field label attribute.
 - **row-number** : User can use this attribute when text is not working, you have to row attribute for providing row number.
 - **submitLabel:**
specifies the label of the button to submit the selected search results. User will have to specify the text seen on the submit button.
- **Example:**

```
// need to update Example and UI image:
```

Created with the Personal Edition of HelpNDoc: [Streamline Your Documentation Process with HelpNDoc's Project Analyzer](#)

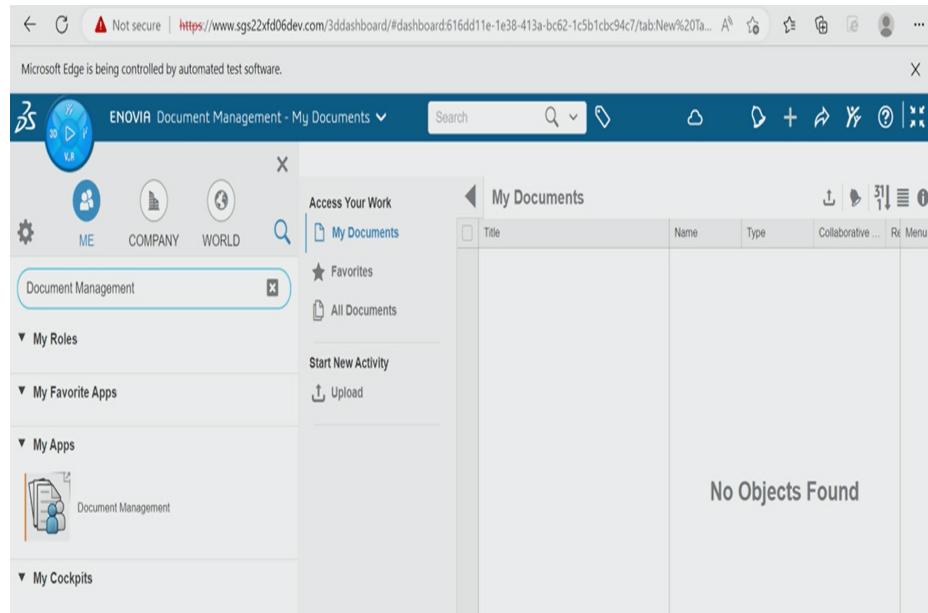
UploadFileForDashboard

UploadFileForDashboard:

Upload file to 3DDashboard using drag and drop method.

- **Syntax:**
`<uploadFileForDashboard path="" locatorType="" locatorExpression="" />`
- **Attribute:**
 - **locator Type: (mandatory)**
There are 4 types of locator: id, cssselector, name and xpath which are already discussed in ClickElement tag.
 - **locatorExpression: (mandatory)**
This attribute contains expression specific to locator type.
 - **path: (mandatory)**
This attribute specifies the file path on local drive to upload it to 3DDashboard.
- **Example**

```
<uploadFileForDashboard path="$csv{input}" locatorType="xpath" locatorExpression="//div[@class='wux-datagridview-emptycontentmessage']//div[contains(text(),'No Objects Found!')]"/>
```



Created with the Personal Edition of HelpNDoc: [Create cross-platform Qt Help files](#)

UploadFileUsingDragAndDrop

UploadFileUsingDragAndDrop:

Upload file to 3DEXperience using drag and drop method.

- **Syntax:**

```
<UploadFileUsingDragAndDrop path="" locatorType="" locatorExpression="" />
```

- **Attribute:**

- **locator Type: (mandatory)**

There are 4 types of locator: id, cssselector, name and xpath which are already discussed in ClickElement tag.

- **locatorExpression:(mandatory)**

This attribute contains expression specific to locator type.

- **path:(mandatory)**

This attribute specifies the file path on local drive to upload it to 3DEXperience.

- **Example:**

Please check below image. Suppose user want to upload file to the Part shown in below image.

Then he can use this tag like

```
<UploadFileUsingDragAndDrop path="D:\Amit\Share\Test.docx"
locatorType="xpath" locatorExpression="//div[@id='divDrag']"/>
```

Properties		Images	Where Used	Revisions	History
<input type="button" value=""/>					
Name	A-0000104-01				
Primary Image					
Part Family					
Design Responsibility	Company Name				
Type	Part				
Revision	A				
Policy	Development Part				
State	Create				
Description					
Change To Release					
Owner	Test Everything				
Originated	30-Mar-2018				
Modified	28-May-2018				
Estimated Cost	0.0 Dollar				

Created with the Personal Edition of HelpNDoc: [Produce online help for Qt applications](#)

UploadFileUsingLocator

UploadFileUsingLocator:

Upload file to 3DEXperience using locator of element which will open file browser dialog. Any of the attribute id/path is mandatory if both attribute used then id will be taken and path will be ignored

Note: This tag is only applicable for `html` tag `input`.
Hence it will only work for `input html field`

- **Syntax:**

`<UploadFileUsingLocator path="" locatorType="" locatorExpression="" id="" />`

- **Attribute:**

- **path: (optional if id used)**

Specifies the path of the file to upload from local file system.

- **id:(optional if path used)**

id attribute is used to take value to search from csv. id attribute is already described in [InputText](#) tag. Please check it.

- **Locator Type: (mandatory)**

There are 4 types of locator: id, cssselector, name and xpath which are already discussed in ClickElement tag.

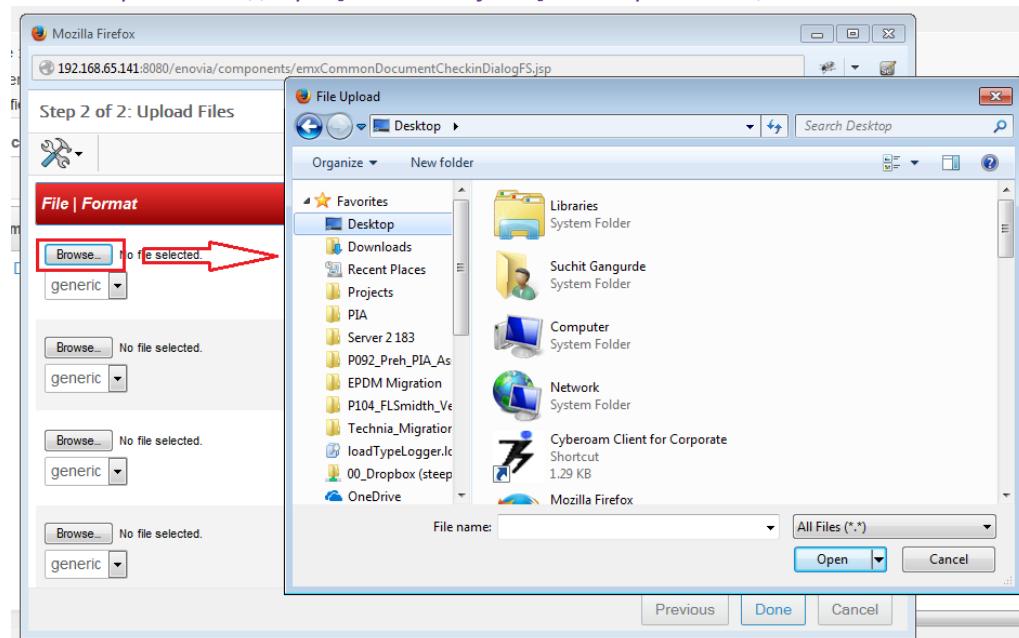
- **locatorExpression:(mandatory)**

This attribute contains expression specific to locator type.

■ **Example:**

In the image below the script is to upload a file to a document. In this case the tag will be written as:

```
<UploadFileUsingLocator path="C:\Users\Desktop\test.txt" locatorType="xpath"
locatorExpression="//input[@name='bfile0']" id="UploadFile"/>
```



Created with the Personal Edition of HelpNDoc: [Say Goodbye to Documentation Headaches with a Help Authoring Tool](#)

ValidateBegin

ValidateBegin :

With the help of this tag,Created new tag for validate the Excel files data with columns number, rows number , sheet number and it's start from 0.

■ **Syntax:**

```
<ValidateBegin filePath=" "/>
```

NOTES:

1. This tag use to validate xls , xlsx files.
2. This tag starts the validation process.

■ **Attributes:**

- **filePath:(mandatory)**

This attribute is Mandatory. Path of the building block (xlsx, xls file) present in testsuites.

■ **Examples:**

Suppose you want to write some skipped testcases

```
<ValidateBegin filePath="C:\01 Ram Ahire\01 TAS\01 A JIRA Selenium testing task\SELENIUM-1221\Reports\BoM Status Report KRD.xlsx"/>
```

```
<ValidateBegin filePath="C:\01 Ram Ahire\01 TAS\01 A JIRA Selenium testing task\SELENIUM-1221\Reports\BoM Status Report KRD.xlsx"/>
```

Created with the Personal Edition of HelpNDoc: [Revolutionize your documentation process with HelpNDoc's online capabilities](#)

ValidateBehaviour

ValidateBehaviour :

With the help of this tag,Get the data for validate the Excel files data with columns number, rows number , sheet number and it's start from 0.

- **Syntax:**

```
<ValidateBehaviour data="" row="" col="" sheet="" />
```

NOTES:

1. This tag use to define the behavior of the data in xls , xlsx file..

- **Attributes:**

- ***data:(Mandatory)***

This attribute is used to validate data from given sheet number, if sheet is empty or null, it will take default sheet 0.

- ***row:(Mandatory)***

This attribute is used to validate row data, only if col is empty or null.

- ***col:(Mandatory)***

This attribute is used to validate col data, only if row is empty or null.

- ***sheet:(Mandatory)***

This attribute is used to validate data from given sheet number, if sheet is empty or null, it will take default sheet 0.

- **Examples:**

Suppose you want to write some skipped testcases.

```
<ValidateBehaviour data="R1F" row="1" col="1" sheet="0" />
<ValidateBehaviour data="Product Number" row="23" col="0" sheet="1" />
```

- **Examples: for better understanding refer image with syntax;the higlited part in image is the input data for validate behaviour**
- **note- in excl column will start from zero, row will start from zero, and sheet also start with zero.**

1	Name - Sheetal Kamthane.		
2	Project Name - 1864 TestAutomationFramework_Selenium.		
3			
4			
5			
6			
7	Task	Date	Hrs
8	Tas Environment set up	13-09-2021	2 Hrs
9	KT for Tas overview	14-09-2021	2.5 Hrs
10	GIT		1 Hr
11	Config in 3 DX Tas		1Hr
12	test case executions		2 Hrs
13	task to creat parts	15-09-2021	2Hrs
14	Script Creation		3Hrs
15	meetings		2 hrs
16			
17	changes in jars	16-09-2021	1Hrs
18	meetings		3 Hrs
19	3Dx tas doc. Reading		2Hrs
20			
21	creating script using tags	19-09-2021	8Hrs
22	creating script using tags	20-09-2021	8Hrs
23	KT for Jira	21-09-2021	1Hrs
24	runing testcases in different browsers		7 hrs
25	creation of jira for failed testcases	22-09-2021	1Hrs
26	writing test cases using different Tgs	22-09-2021	5Hrs
27			
28	writing test cases using different Tgs	23-09-2021	8Hrs
29	writing test cases using different Tgs	24-09-2021	8Hrs
30			
31	writing test cases using different Tgs	26-09-2021	8Hrs
32			
33			
34			
35			
36			
37			
38			

```
<ValidateBegin filePath="D:\1864 Project daily worksheet.xlsx"/>
  <ValidateBehaviour data="Tas Environment set up" row="7" col="0" sheet="0"/>
<ValidateEnd/>
```

Created with the Personal Edition of HelpNDoc: [Effortlessly create a professional-quality documentation website with HelpNDoc](#)

ValidateEnd

ValidateEnd :

With the help of this tag, End tag of data validation of Excel files. last tag of validation which comes after ValidateBehaviour.

- **Syntax:**

```
<ValidateEnd/>
```

NOTES:

1. This tag use to ending the process of data validation of the data in xls , xlsx file.
2. Only read xls and xlsx files, other wise it will give Error.

- **Example:**

```
<ValidateBegin filePath="testsuites\SELENIUM-1440\BOM Status Report 1.xlsx"/>
  <ValidateBehaviour data="Bom Status Report Input" />
<ValidateEnd/>
```

Created with the Personal Edition of HelpNDoc: [Benefits of a Help Authoring Tool](#)

ValidateTableExport

ValidateTableExport :

This tag allows the user to validate the table export in CSV format and ensure the table exists in the UI. It exports the CSV file and compares the downloaded CSV table with the table in the UI for validation.

- **Syntax:**

```
<ValidateTableExport headerList="" skipCount="3" filePath="" time="" />
```

NOTES:

1. This tag exports the CSV file and compares the downloaded CSV table with the table in the UI for validation.
2. In this tag user should give proper file path and headrList which you want to validate into the downloaded csv file.
3. This tag works only if the freeze pane of the table contains only a single column with data in the form of a hyperlink. For non-freeze panes, it can function with or without a hyperlink.
4. All table headers must have a unique name.
5. This tag cannot validate table headers having icons or images or blank headers.
6. This tag can work with the table data having icon/images/blank data.

- **Attributes:**

- **headerList:(Mandatory)**

If not provided header list from the csv file headerName then it will give you Error. so please give correct header list name with Pipe separated value. list of headers to be matched in csv and UI table.

- **skipCount:(Mandatory)**

If not provided Skip column count number from the csv file then it will give you Error. so please give skipCount column number regarding empty column. if column is not empty then provide 0 value in skipCount="0", Number of lines to be skipped in csv file before reading header.

- **filePath:(optional)**

please provide specified file path name, file path in case of modified download path in browser. By default, the path will be taken from the key 'sg-tas.download.filepath' in the SG_TAS.properties file.

- **time:(optional)**

This attribute contains wait time in milliseconds to find element. Default value will be taken from '3dx-tas.execution.step.timeout'. Preferred wait is 5000MS..

- **skipError:(optional)**

Possible values are true and false. default value is false to throw the error and true to ignore the error.

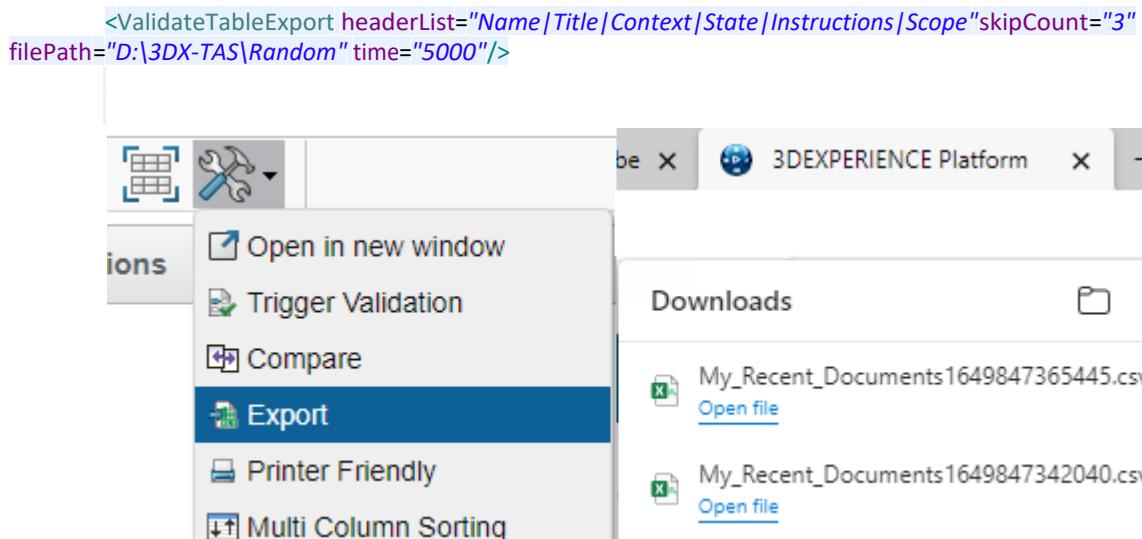
- **refIdForTool:(optional)**

This attribute is optional and it will be used only when in UI user get multiple elements for

'//div[@id='divToolbar']//td[@title='Tools']/img[@src='..//common/images/iconSmallAdministration.png']' this xpath & specific xpath will be required to run this tag. In this case user can use the id attribute of findElement and refer the ID here as refIdForTool..

■ **Examples:**

Suppose you want to validate header list from CSV file.



Below is the example for pre-requisite of this tag that this tag works only if the freeze pane of the table contains only a single column with data in the form of a hyperlink. For non-freeze panes, it can function with or without a hyperlink.

The screenshot shows the 3DEXPERIENCE Platform Dashboard. The main area displays a table with columns: Name, Title, Rev, Ver, Type, Mod, Stat, and Description. A specific cell in the 'Title' column for row 10 ('DOC-0000432') contains a hyperlink labeled 'DOC-0000431'. The table has a frozen header row. At the bottom of the table, a message reads 'Documents that have been created within the last 10 ...' and '100 objects'.

ValidateTableHeader

ValidateTableHeader :

With the help of this tag, user should be able to Validate table header in downloaded file.

- **Syntax:**

```
<ValidateTableHeader headerList="" downloadFileExt="" skipCount="" filePath="" />
```

NOTES:

1. This tag is use to validate only when downloaded file is csv,xlsx or xls.
2. In this tag user should give proper Extension.

- **Attributes:**

- **headerList:(Mandatory)**

If not provided header list from the downloaded file headerName then it will give you Error. so please give correct header list name with Pipe separated value. list of headers to be matched in any file and table. The sequence of header list should be same as downloaded file headers.

- **downloadFileExt:(Optional)**

Extension of the file. If not provided Extension then it will by default take csv Extension. Possible values are csv,xlsx or xls.

- **skipCount:(Optional)**

Number of lines to be skipped in downloaded file before reading header.

- **filePath:(optional)**

filePath is used to specify directory of downloaded file.

- **Examples:**

Suppose you want to validate header list from CSV file.

```
<ValidateTableHeader headerList="username/Password" downloadFileExt="csv" skipCount="2"
filePath="C:\TAS_Ericsson\Downloads"/>
```

Created with the Personal Edition of HelpNDoc: Easily create Help documents

Wait

Wait:

Pause the execution of test case for given time. In other words, wait for something to happen like wait for page to load before performing any action.

Most of the web applications are developed using Ajax and JavaScript. When a page is loaded by the browser the elements which we want to interact with may load at different time intervals.

Not only it makes this difficult to identify the element but also if the element is not located it will throw an ElementNotFoundException exception. Using Waits, we can resolve this problem.

- **Syntax:**

```
<Wait time="1000" />
```

- **Attribute:**

- **time:(mandatory)**

The time attribute defines wait interval. User must pass time in milliseconds.

- **afterTime:(optional)**

possible value - pass , fail. whether to pass or fail the test case if element not found

- **for:(optional)**

This is an optional attribute. The possible values of this attribute are time (like 2000), page (i.e. it would wait for page to load for given time (attribute) value) and element. The time attribute value act as a threshold value for page to load completely.

- **locatorType:(mandatory only for for="element")**

This attribute is already defined for ClickElement tag. User could refer to same.

- **locatorExpression:(mandatory only for for="element")**

This attribute is already defined for ClickElement tag. User could refer to same.

NOTE:

1. locatorType and locatorExpression are used only when value of for="element". It is used to implement explicit wait. It is useful when you need to wait for a particular element to be available before moving to the next step.
2. e.g if the mentioned wait time is 5 seconds and the element is available at 3 seconds then the WebDriver will not wait beyond 3 seconds as the element is available.
3. If the condition is not met within the specified timeout period, the WebDriver throws a TimeoutException.

■ **Example:**

```
<Wait for="page" time="10000" />
```

In the above example the maximum time it would wait for page to load completely is 10 seconds.

```
<Wait for="element" time="10000" locatorType="xpath"
locatorExpression="//h2[@title='Product Lines']" />
```

In the above example, the maximum time it would wait for element to appear is 10 seconds.

Created with the Personal Edition of HelpNDoc: [Easily create PDF Help documents](#)

RefreshCurrentPage

RefreshCurrentPage:

User can use RefreshCurrentPage tag to refresh the current page.

■ **Syntax:**

```
<refreshCurrentPage />
```

■ **Example:**

if you want to refresh the current page, this tag can be used.

```
<refreshCurrentPage />
```

The screenshot shows the 3DEXPERIENCE | ENOVIA Collaboration and Approvals web application. The left sidebar includes sections for Change Management (My Changes), Collaboration and Approvals (Dashboard, Issues Summary, Collections), Utilities (Background Jobs, About...), and a download link for Microsoft Client. The main area displays a 'Dashboard' with tabs for New Docs, Changed Docs, My Recent Docs, and My Trainings. A table lists documents with columns for Name, Title, Actions, Rev, Ver, and a small icon. On the right, there are panels for Assigned Items and Updates of documents, along with a progress bar.

Created with the Personal Edition of HelpNDoc: [Experience the Power and Ease of Use of a Help Authoring Tool](#)

Native

This custom xml tag library is defined for SteepGraph test automation tool(3DX Native App Recorder). Each tag has a specific action to perform on desktop/native application. With help of this tool, user could easily perform different actions like Clicking different elements, input text using SendKeys etc.

Guidelines:

1. While sending keystrokes through sendKeys, user should enter the text at normal speed[not in fast speed].
2. While Recording script, user should move the cursor at normal speed, so that the actions are recorded properly.
3. While creating Building Block, the images folder should be created under Building Blocks folder for placing the images captured for scripts. This is applicable for general Tags. For 3DX tags, the images should be replaced under common images folder.

The details of each tag is defined in below sub-topics/sub-modules:

Created with the Personal Edition of HelpNDoc: [Maximize Your Productivity with a Help Authoring Tool](#)

3DX Tags

1. Login:

With the help of this tag, user should be able to login to 3DX Native Apps, by entering username, password and platformName attributes. This could be used when we need to change user in between test case. Before using login tag, we need to logout from 3DX Native Application.

- **Syntax:**

```
<Login username="" password="" platformName="" />
```

NOTES:

1. User should be logged out, in order to login into 3DX Native Application.
2. The images captured for locatorImage attribute should always be in .png extension. The image captured for the required element should be focused and captured properly.
3. In Login tag, both username and password are mandatory attributes. platformName attribute is optional.
4. User should capture image of ENTER_USER_NAME field [present in common images folder]. There are few images which is present in common images folder for

3DX Tag attribute. User may or may not need to replace those images depending upon their screen resolution], having cursor on the field as provided below:



ENTER_USER_NAME

- **Attributes:**

- **username:**

This attribute refers to the valid username which needs to be entered in the username field, in order to login to 3DX Native Application.

- **password:**

This attribute refers to the valid password which needs to be entered in the password field, in order to login to 3DX Native Application.

- **platformName:**

This attribute refers to the valid platformName which needs to be entered in the platformName field, in order to login to 3DX Native Application.

- **Example:**

<Login username="3dx_user2" password="Passport1" />

2. OpenCompassApp:

With the help of this tag, user could click any of the given Compass Quadrant and open the any of the required Compass App from it by using quadrant, appnameimage and appname attributes. User must provide quadrant, appnameimage and appname in order to open the application.

- **Syntax:**

<OpenCompassApp quadrant="" appnameimage= appname="" />

NOTES:

1. The images captured for locatorImage attribute should always be in .png extension. The image captured for the required element should be focused and captured properly.
2. The quadrant, appnameimage and appname are mandatory attributes.

- **Attributes:**

- **quadrant:**

This attribute refers to the quadrant, from which user needs to launch the required application. The possible values are north, south, east and west.

- **appnameimage:**

This attribute refers to the specific image of appname, which user need to launched.

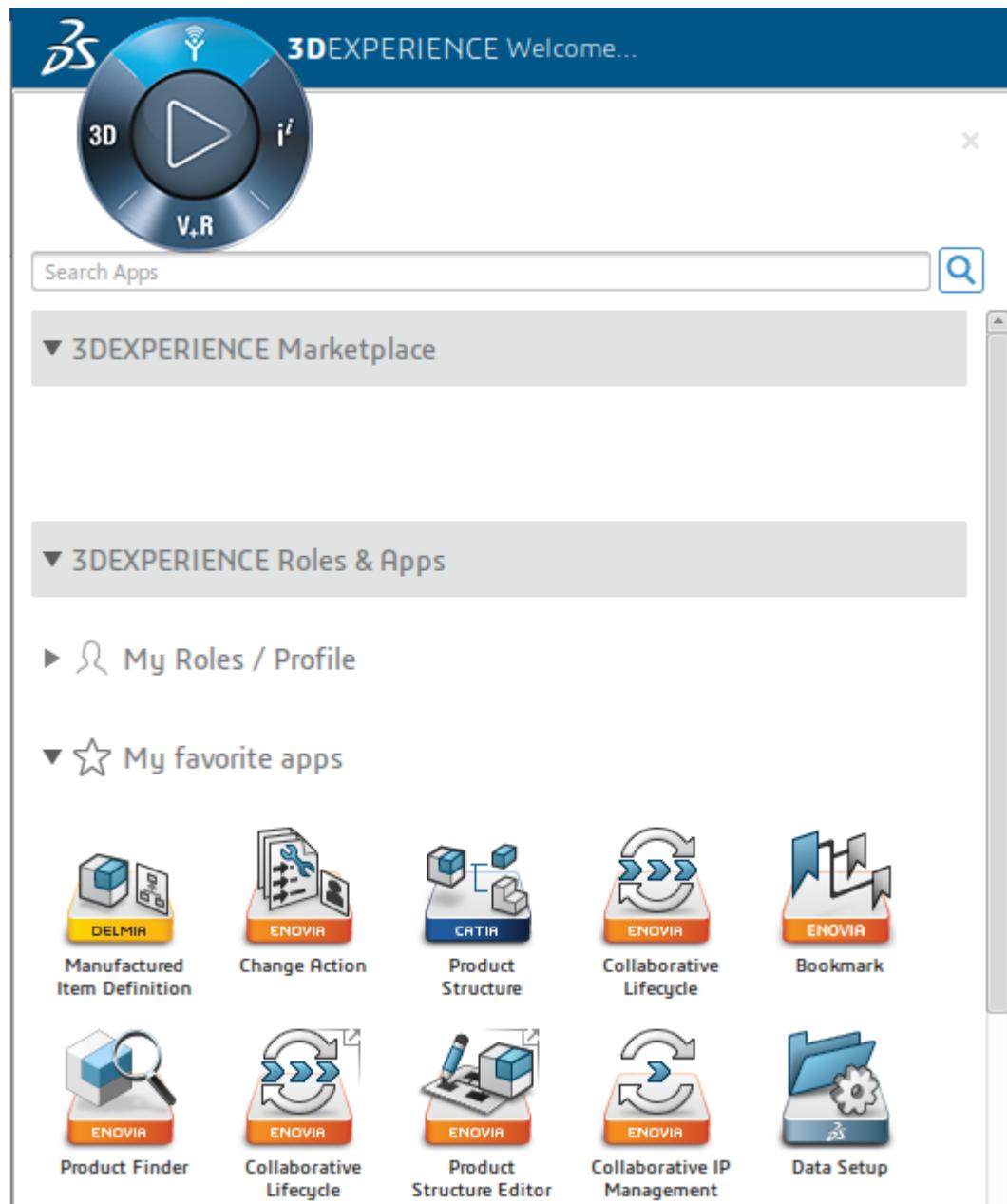
- **appname:**

This attribute refers to the specific text of the appname, on which user needs to click, in order to launch the application.

- **Example:**

In the image below, the user wants to open Product Structure App from north quadrant. The tag in this case will be as follows:

<OpenCompassApp quadrant="north" appnameimage="1556778638.png" appname="Product Structure"/>



3. Global Search:

With the help of this tag, user should be able to search object in 3DX Native Application.

- **Syntax:**

```
<GlobalSearch id="" type="" searchtype="">
```

NOTES:

1. The id attribute is mandatory and rest of the attributes(i.e. type and searchtype are optional).

- **Attributes:**

- **id:**

This attribute refers to the text, which needs to be searched by the user and the same should be provided in csv. This attribute id is mapped to csv column header. User should provide the column header in the script, wherever, he needs to provide the reference of this.

- **type:**

This attribute refers to type from global search type. The default type value for this attribute is All.

- **searchtype:**

This attribute refers to search type, whether user wants to perform normal search or Advanced search. The possible values for this attribute is either Search or Advance Search. The default searchtype is Search.

- **Example:**

Support user like to search A-0000104-01 Part in 3DEXperience. First, he/she need to define column in csv and enter its value as "A-0000104-01".

```
"PartName", "Qty"  
"A-0000104-01", "1"
```

Then define tag in test case like

```
<GlobalSearch id="PartName" type="All" searchtype="Search"/>
```

Then you will see the global search result like

Display Name	Revision	Type	Description	State	Modified
A-0000104-01	A	Part	Create	Create	28-May-2018

4. RegisterObject :

With the help of this tag, user should be able to register an object's name in a property file or in database by given locatorImage, id and elements co-ordinates(i.e. l,r,t and b) attributes.

To determine the destination in which the object's name will be registered, they must exist in TestAutomationFramework.properties file the following keys:

- **object.registration.type:** Specifies where to register the object name. It can take two values, either propertyfile or database.
E.g. object.registration.type=propertyfile
- **object.registration.filepath:** Specifies the name of the file and path in your file system where it will be located.

E.g. object.registration.filepath=src\main\resources\objectregister.properties

- **Syntax:**

`<RegisterObject locatorImage= id="" l="" r="" t="" b="" />`

NOTES:

1. The images captured for locatorImage attribute should always be in .png extension. The image captured for the required element should be focused and captured properly.
2. In RegisterObject tag, locatorImage and id are mandatory attributes and rest all attributes are optional(i.e. elements co-ordinates).

- **Attributes:**

- **locatorImage:**

This attribute refers to specific image of an element, which needs to be registered.

- **id:**

This attribute refers to the Key, used to register the data.

- **l(left):**

This attribute refers to the left co-ordinates of the elements, which needs to be registered. This needs to be entered by the user manually.

- **r(right):**

This attribute refers to the right co-ordinates of the elements, which needs to be registered. This needs to be entered by the user manually.

- **t(top):**

This attribute refers to the top co-ordinates of the elements, which needs to be registered. This needs to be entered by the user manually.

- **b(bottom):**

This attribute refers to the bottom co-ordinates of the elements, which needs to be registered. This needs to be entered by the user manually.

- **Examples:**

In the images below, the user wants to register the name of the Part in a property file. The tag in this case will be:

`<RegisterObject locatorImage="12255667433.png" id="Part_Name" />`

In the csv file, a header with the name `Part_Name` must exist whose value will be the key with which the object will be registered.

```
orA-0000150-01,Test,1,eForAddExisting,Estimated_Start,"Filter_Person" "Part_Name"
,"Registered_Part"
```

5. launchapplication:

With the help of this tag, user should be able to launch the required application using command attribute. It could either be mentioned by the user manually or through csv.

- **Syntax:**

```
<LaunchApplication command="" />
```

- **Attributes:**

- **command:**

This attribute refers to application path or command , in order to launch application.

- **Examples:**

For e.g.: Suppose user wants to launch 3DX Native application, the syntax for the same is provided below:

```
<LaunchApplication command="C:\Program Files\Dassault
Systemes\B421\win_b64\code\bin\CATSTART.exe" />
```

```
<LaunchApplication command="C:\Program Files\Dassault
Systemes\B421\win_b64\code\bin\CATSTART.exe" -run "3DEXPERIENCE" -env
Env -direnv "E:\Work\STRATA\STR_DELMIA_Commands" -nowindow />
```

NOTE: When user enters have to enter filepath from Excel sheet the filepath should be written with "/"

Created with the Personal Edition of HelpNDoc: [Easily create PDF Help documents](#)

General Tags

1. Click Element:

With help of this tag, user could identify Element which needs to be clicked, using given locatorImage and element coordinates(i.e. l(left), r(right), b(bottom) and t(top)). User can also mention locatorText in script itself, once the script is imported into eclipse, for correct

identification of element,

- **Syntax:**

```
<ClickElement locatorImage="" l="" r="" t="" b="" />
```

NOTES:

1. The images captured for locatorImage attribute should always be in .png extension. The image captured for the required element should be focused and captured properly.
2. In ClickElement tag, locatorImage are mandatory attribute and rest all attributes are optional(i.e. element co-ordinates(i.e. l,r,t,b)).

- **Attributes:**

- **locatorImage:**

This attribute refers to image captured, specific to the element, which needs to be clicked by the user. The image captured should always have .png extension

- **l(left):**

This attribute refers to the left co-ordinates of the elements, which needs to be clicked. This needs to be entered by the user manually.

- **r(right):**

This attribute refers to the right co-ordinates of the elements, which needs to be clicked. This needs to be entered by the user manually.

- **t(top):**

This attribute refers to the top co-ordinates of the elements, which needs to be clicked. This needs to be entered by the user manually.

- **b(bottom):**

This attribute refers to the bottom co-ordinates of the elements, which needs to be clicked. This needs to be entered by the user manually.

- **Example:**



ENTER_USER_NAME

Syntax:

```
<ClickElement locatorImage="125566688788.png" l="-1" r="0" t="0" b="-2" />
```



Physical_Product

Syntax:

```
<ClickElement locatorImage="125566333788.png" />
```

Work Offline

Work1

Syntax:

```
<ClickElement locatorImage="Work1.png" locatorText="Work Offline" />
```

2. ***FindElement:***

With help of this tag, user could find Element which needs to be searched, using given locatorImage, id, text and element coordinates(i.e. l(left), r(right), b(bottom) and t(top)).

- **Syntax:**

```
<FindElement id="" locatorImage="" text="" l="" r="" t="" b="" />
```

NOTES:

1. The images captured for locatorImage attribute should always be in .png extension. The image captured for the required element should be focused and captured properly.
2. In FindElement tag, locatorImage and id are mandatory attribute and rest all attributes are optional(i.e. text and element co-ordinates(i.e. l,r,t,b)).

- **Attributes:**

- **locatorImage:**

This attribute refers to image captured, specific to the element, which needs to be searched by the user. The image captured should always have .png extension.

- **id:**

This attribute refers to the reference id of Image available in locatorImage attribute. User could identify element for once, with FindElement tag and re-use it again for the other tags with the help of id attribute.

- **text:**

This attribute refers to the text value, present on image.

- **l(left):**

This attribute refers to the left co-ordinates of the elements, which needs to be clicked. This needs to be entered by the user manually.

- **r(right):**

This attribute refers to the right co-ordinates of the elements, which needs to be clicked. This needs to be entered by the user manually.

- **t(top):**

This attribute refers to the top co-ordinates of the elements, which needs to be clicked. This needs to be entered by the user manually.

- **b(bottom):**

This attribute refers to the bottom co-ordinates of the elements, which needs to be clicked. This needs to be entered by the user manually.

- **Example:**

ENTER_USER_NAME

Syntax:

```
<FindElement id="username" locatorImage="125566688788.png" text="Username" l="-1" r="0" t="0" b="-2"/>
```

3. ***inputtext:***

With help of this tag, user could insert a text into input field (element) identified using given locatorImage, input and element co-ordinates(i.e. l,r,t,b) attribute. Input value is taken from input attribute.

NOTES:

1. The images captured for locatorImage attribute should always be in .png extension.
The image captured for the required element should be focused and captured properly.
2. In inputtext tag, locatorImage and input are mandatory attributes and rest all attributes are optional(i.e. element co-ordinates(i.e. l,r,t,b)).

- **Syntax:**

`<InputText locatorImage="" input="" l="" r="" t="" b="" />`

- **Attributes:**

- **locatorImage:**

This attribute refers to image captured, specific to the element(textfield), for which user needs to provide input/text. The image captured should always have .png extension

- **input:**

This attribute refers to the text which needs to be entered by the user in textfield, for Image available in locatorImage attribute.

- **l(left):**

This attribute refers to the left co-ordinates of the elements, for which text needs to be entered. This needs to be entered by the user manually.

- **r(right):**

This attribute refers to the right co-ordinates of the elements, for which text needs to be entered. This needs to be entered by the user manually.

- **t(top):**

This attribute refers to the top co-ordinates of the elements, for which text needs to be entered. This needs to be entered by the user manually.

- **b(bottom):**

This attribute refers to the bottom co-ordinates of the elements, for which text needs to be entered. This needs to be entered by the user manually.

- **Example:**



ENTER_USER_NAME

Syntax:

`<InputText locatorImage="125566688788.png" input="3dx_user2" l="-1" r="0" t="0" b="-2"/>`

4. ***SendKeys:***

With help of this tag, user could enter text in form of keystrokes sent, to an element in an application using given key attribute.

For example: Suppose user wants to enter some text in Password field. Firstly user clicks on Password field and would enter text(i.e. user could send keystrokes(i.e.required text) to the an element(i.e. Password field)).

- **Syntax:**
`<SendKeys key="">`
- **Attributes:**
 - **key:**
This attribute refers to text which needs to be entered by user for an element, in form of keystrokes.
- **Example:**



Syntax:
`<ClickElement locatorImage="125569878788.png"/>`
`<SendKeys key="Pass1"/>`

5. Wait:

With the help of this tag, user could pause the execution of test case for given time. Using Waits, we could resolve multiple issues which might occur in our application while testing like Element didn't appeared due to more time taken by screen to load or screen was not loaded properly etc.

- **Syntax:**
`<Wait time="">`
- **Attributes:**
 - **time:**
The time attribute defines wait interval. User must pass time in milliseconds.
- **Example:**
`<Wait time="1000"/>`

6. Action:

With the help of this tag, user could perform a given action on element such as click, doubleclick, rightclick etc. by using refid, name, locatorImage and value attributes, but to achieve that user first needs to locate element using FindElement first.

- **Syntax:**
`<Action refId="" name="" locatorImage="" value="">`

NOTES:

1. The images captured for locatorImage attribute should always be in .png extension. The image captured for the required element should be focused and captured properly.
2. In Action tag, refId and name are mandatory attributes and rest all attributes are optional(i.e. locatorImage and value).

- **Attributes:**
 - **refid:**

This attribute refers to the refid, which user needs to get from findelement tag.

- ***name:***

This attribute refers to Action name, user wants to perform such as click, doubleclick, rightclick, keyup, keydown, input, mousedown, mouseup, mousemove etc.

- ***locatorImage:***

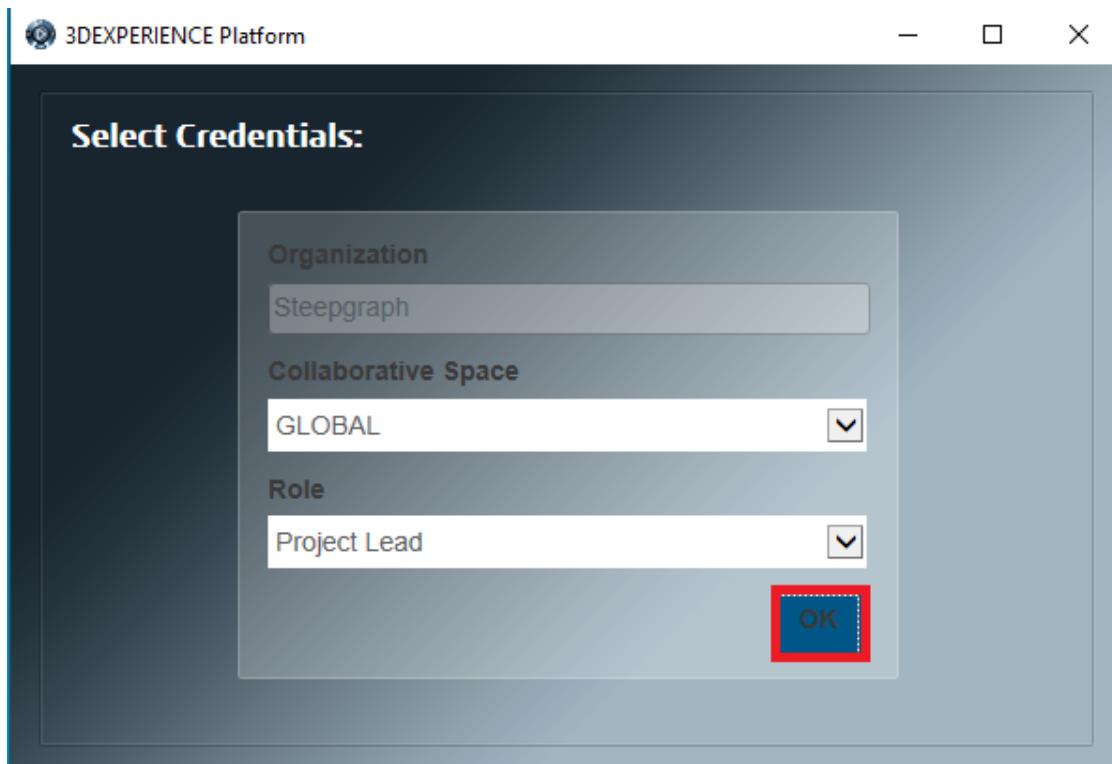
This attribute refers to specific image of an element, on which user wants to perform an action. This attribute is optional and could be used, in case refid is not given.

- ***value:***

This attribute refers to text/modifier keys (Key.ALT, Key.CMD, Key.CTRL, Key.META, Key.SHIFT, Key.WIN, Key.ALTGR) and is optional.

- ***Example:***

When user wants to perform click on an element. Functionally action tag and ClickElement are similar, but technically they are different. For some cases ClickElement tag is not able click on element, in such cases use action tag. Suppose we want to click on OK button shown in below image:



In order click on OK button, we need to define tags in test case xml as following:

```
<FindElement id="Okbtn" locatorImage="1343335677.png" text="OK"/>
<Action refid="Okbtn" name="click" />
```

7. Assert Tag:

With the help of this tag, user could assert values for validation. The Assertion results are based on the comparison of Actual & Expected Results.

- ***Syntax:***

```
<Assert criteria="" locatorImage="" timeout="" textToValidate="" colorDifference=""
```

I="" r="" t="" b=""/>

NOTES:

1. The images captured for locatorImage attribute should always be in .png extension. The image captured for the required element should be focused and captured properly.
2. In Assert tag, criteria, locatorImage and timeout are mandatory attributes and rest all attributes are optional(i.e.element co-ordinates). For remaining tag(i.e. text and colorDifference) it would depend on the assertion criteria.
3. In case user wants to record the Assert Tag with help of Record mode, user should hold right click and press key "a"(Right click + a), while holding right click and then release.

- **Attributes:**

- **criteria:**

This attribute refers to the value/criteria, based on which the Actual and Expected values are compared. The possible values for this could be image, text or color.

- **locatorImage:**

This attribute refers to specific image of an element, which needs to be validated based on the assertion applied.

- **timeout:**

This refers to the time limit provided by the user to check assertion applied.

- **text:**

This attribute refers to the expected value of text which needs to be validated and it would be used if and only if criteria value is specified as text.

- **colorDifference:**

This attribute refers to the expected value of color which needs to be validated and it would be used if and only if criteria value is specified as color.

The possible values of colorDifference could be 0.07,0.7,0 etc.

- **I(left):**

This attribute refers to the left co-ordinates of the elements, which is specified in locatorImage. This needs to be entered by the user manually.

- **r(right):**

This attribute refers to the right co-ordinates of the elements, which is specified in locatorImage. This needs to be entered by the user manually.

- **t(top):**

This attribute refers to the top co-ordinates of the elements, which is specified in locatorImage. This needs to be entered by the user manually.

- **b(bottom):**

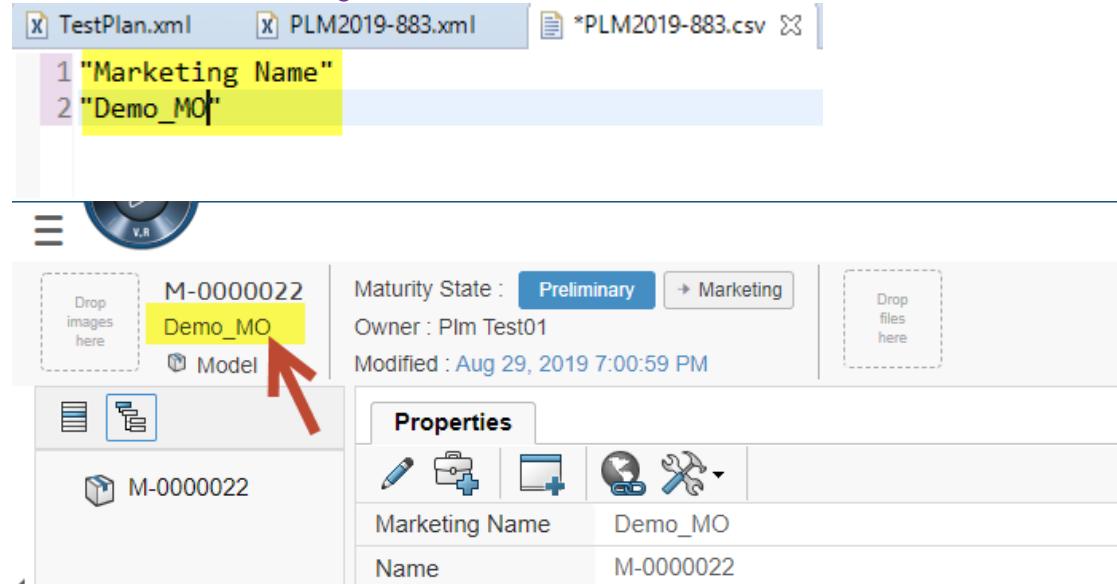
This attribute refers to the bottom co-ordinates of the elements, which is specified in locatorImage. This needs to be entered by the user manually.

- **Examples:**

In the image below the user wants to validate the created object. Assert will be compared element text with csv input file(Value of id="Marketing Name" column)

<Assert criteria="Text" locatorImage="576544564564.png" timeout="20"

textToValidate="Marketing Name"/>



1587137907601.png

```
<Assert criteria="color" locatorImage="1587137907601.png" timeout="2000"
colorDifference="0.07"/>
```

8. FileChooser:

With the help of this tag, user should be able to open specified file. User could also open multiple files, if filepath remains same for all the files. User could enter multiple filenames separated by pipe '|' symbol.

For e.g.: filenames="Part.txt|Requirement.txt"

- **Syntax:**

```
<FileChooser locatorImage="" filenames="" filepath="" />
```

NOTES:

1. The images captured for locatorImage attribute should always be in .png extension. The image captured for the required element should be focused and captured properly.
2. In FileChooser tag, all the three attributes are mandatory.

- **Attributes:**

- **filenames:**

This attribute refers to the specified filename/filenames(along with their extension), which needs to be provided by the user, in order to open the required file's.

- **locatorImage:**

This attribute refers to specific image from where user needs to choose the file.

- **filepath:**

This refers to the path, where files are placed, which user wants to open.

- **Examples:**

For e.g. : Suppose user wants to open a file named "1023.txt" and "1023_1.txt" from

the image provided below:

	Name	Date modified	Type	Size
Quick access	1	31-01-2020 16:17	Microsoft Word D...	1,648 KB
Desktop	1023	24-01-2020 14:13	File	2 KB
Downloads	1023_1	24-01-2020 15:00	File	4 KB

The Syntax for the same is provided below:

```
<FileChooser locatorImage="3432432533.png" filenames="1023.txt|1023_1.txt"
filepath="E:\Work\|"/>
```

9. IF/ELSEIF/ELSE/ENDIF:

With the help of this tag, user should be able to check condition(i.e. based on text comparison and visibility). One Script may/may not have multiple IF/ELSEIF statements. If user wants to check single condition he/she could use IF Tag. In case, wants to check for multiple conditions for different cases, he/she could use IF Tag combined with ELSEIF Tag.

- **Syntax:**

```
<IF id="" criteria="" condition="" locatorImage="" refId="" wait="" />
```

NOTES:

1. The images captured for locatorImage attribute should always be in .png extension. The image captured for the required element should be focused and captured properly.
2. In ifCondition tag, the mandatory attributes are id, criteria, condition, locatorImage and refId.

- **Attributes:**

- **id:**

This attribute refers to the text value of the element, which user need to compare.

- **criteria:**

This attribute refers to the specified condition, which needs to be checked. The possible values of the criteria are Visible or Text.

- **condition:**

This attribute refers to the conditional operator, based on which condition would be verified.

- **refId:**

This refers to the refId, which user needs to get from findelement tag or else locatorImage of the element..

- **locatorImage:**

This attribute refers to specific image of the element for which user needs to verify the condition.

- **wait:**

This attribute refers to time attribute defines wait interval. User must pass time in milliseconds.

- **Examples:**

Suppose user wants to compare the username text, the syntax for the same is provided below:

```
<IF id="3dx_user2" criteria="Text" condition="=" locatorImage="1234882773.png"
refId="username" wait="2000"/>
<ClickElement locatorImage="125566677688.png"/>
<ELSE/>
<InputText locatorImage="125566688788.png" input="3dx_user2"/>
<ENDIF/>
```

In the above scenario, user is trying to compare the text in username field, if available. So firstly the if condition is checked, in case the if condition is satisfied, it will click on Password field and the code exits from IFELSE and would be navigated to next line of code after IFELSE. In case, user finds out that there is no text present in Username field and If condition is not satisfied, the the line of code would navigate to ELSE condition and the username would get entered as "3dx_user2" and the code would navigate to the next line of code.

Examples for criteria equal to text and visible is provided below:



i) <IF id="\$csv{Text2}" criteria="Text" condition="=" locatorImage="Global.png"
wait="2000"/>
<ClickElement locatorImage="1585789691189.png" />
<ClickElement locatorImage="1585815035560.png" />
<ELSEIF id="\$csv{Text}" criteria="Text" condition="="
locatorImage="Global.png" />
<ClickElement locatorImage="1585789691189.png"/>
<ELSE/>
<ClickElement locatorImage="icon.png" />



ii) <IF id="inputtext" criteria="visible" condition="="
locatorImage="user.png"/>
<ClickElement locatorImage="user.png"/>
<ELSEIF id="inputtext" criteria="visible" condition="!="
locatorImage="user.png"/>
<ClickElement locatorImage="welcome.png"/>
<ELSE/>
<ClickElement locatorImage="ds.png"/>
<ENDIF />

10. Scroll:

With the help of this tag, user should be able to perform scrolling, as per given inputs. Suppose, if there is any need to scroll or some condition, either scrolling is required(i.e. could be used to make element visible), using given attributes such as steps, locatorImage, direction, stepDelay and type attributes.

- **Syntax:**

```
<Scroll steps="" locatorImage="" direction="" type="" stepDelay="" />
```

NOTES:

1. The images captured for locatorImage attribute should always be in .png extension. The image captured for the required element should be focused and captured properly.
2. In scroll tag, the steps, locatorImage and direction are mandatory attribute and rest all attributes are optional(i.e. stepDelay and type)

▪ **Attributes:**

- **steps:**

This attribute specifies the number of steps/count required for scroll to perform.

- **locatorImage:**

This attribute refers to specific image, which specifies from where scroll should start.

- **direction:**

This attribute refers to the scroll direction. The default value for this attribute is down. The possible values for this attribute are up and down

- **stepDelay:**

This attribute refers to the delay time in milliseconds between two scroll steps. The default value for this attribute is 50.

- **type:**

This attribute refers to the scroll type. The default value for this attribute is wheel. The possible values for this attribute are wheel and key.

▪ **Examples:**

Sometimes element on which user want to perform action is not visible on screen. If we directly tried to perform some action like click, then it will fail. For such cases user need to use this tag to scroll parent scroll bar to make that web element visible on screen.

Please check below image. So, to edit quantity of row, we first must move scroll bar to right. For indented table give xpath of required of first row.

`<Scroll steps="4" locatorImage="scroll.png" direction="down"/>`



scroll.png

11. selectcheckbox:

With the help of this tag, user should be able to select a checkbox from multiple checkboxes or multiple checkboxes for any given region.

- **Syntax:**

```
<SelectCheckBox checkboxregionimage="" checkboxtext="" checkboximage= l="" r="" t="" b="" />
```

NOTES:

1. The images captured for locatorImage attribute should always be in .png extension. The image captured for the required element should be focused and captured properly.
2. In selectcheckbox tag, both checkboxregionImage, checkboxtext and checkboximage attributes are mandatory and rest all attributes are optional(i.e. element co-ordinates)

- **Attributes:**

- **checkboxregionimage:**

This attribute refers to specific region of element/checkbox, from which user wants to select the checkbox.

- **checkboxtext:**

This attribute refers to text value of checkbox , which user wants to select.

- **checkboximage:**

This attribute refers to checkbox image , which user wants to select.

- **l(left):**

This attribute refers to the left co-ordinates of the elements, which is specified in locatorImage. This needs to be entered by the user manually.

- **r(right):**

This attribute refers to the right co-ordinates of the elements, which is specified in locatorImage. This needs to be entered by the user manually.

- **t(top):**

This attribute refers to the top co-ordinates of the elements,which is specified in locatorImage. This needs to be entered by the user manually.

- **b(bottom):**

This attribute refers to the bottom co-ordinates of the elements, which is specified in locatorImage. This needs to be entered by the user manually.

- **Examples:**

Suppose user wants to select one of the checkbox from Native application as provided in image below, the image and syntax for the same is provided below:

Generate Manufacturing Assembly Based on Make-Buy

Create Complete Manufacturing Assembly for Assembly

Syntax:

```
<SelectCheckBox checkboxregionimage="$csv{CheckBoxRegion}"
```

```
checkboxtext="$csv{createCompleteManufacturing}" l="200" r="10" t="20" b="20"/>
```

12. Include:

With the help of this tag, user could include the created building blocks to scripts.

- ***Building Block:***

1. Each building block consists of simple xml script.
2. All the building blocks will be placed in an test suites folder, We can create the separate folder for building blocks E.g: BB
3. These will be later used to create final test script
4. We can use multiple building blocks in single test script

NOTE:

The images for the Building Block should be stored under Images folder, which is present under Building Block folder. In case user is not able to find the same, user should create one and store images in same.

Building Block Example:

Below is the example of building block to promote object.

Login.xml

```

1 <?xml version="1.0" encoding="UTF-8" standalone="no" ?>
2
3 <!-- 3DX Native App Recorder -->
4 <!-- PLEASE WRITE DESCRIPTION OF THE SCRIPT -->
5
6<TestCase xmlns="https://www.steepgraph.com"
7   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
8   xsi:schemaLocation="https://www.steepgraph.com ../../src/main/resources/TestAutomationFrameworkDesktop.xsd"
9   targetApplication="desktop">
10
11  <LaunchApplication
12    filepath="C:\Program Files\Dassault Systemes\B421\win_b64\code\bin\CATSTART.exe" />
13
14  <!-- TBD : Step description -->
15<ClickElement locatorImage="1586509297049.png" />
16
17  <!-- TBD : Step description -->
18<SendKeys key="3dx_user2" />
19
20<!-- TBD : Step description -->
21<ClickElement locatorImage="1586509341835.png" />
22
23<!-- TBD : Step description -->
24<SendKeys key="Passport1" />
25
26<!-- TBD : Step description -->
27<ClickElement locatorImage="1586509371118.png" />
28
29 </TestCase>

```

- ***Syntax:***

```
<Include filepath="testsuites\BB\Login.xml"/>
```

- ***Attributes:***

- ***filepath:***

Path of the building block (xml file) present in testsuites.

- ***Examples:***

To Create Login:

```

1  <?xml version="1.0" encoding="UTF-8" standalone="no" ?>
2
3  <!-- 3DX Native App Recorder -->
4  <!-- PLEASE WRITE DESCRIPTION OF THE SCRIPT -->
5
6@<TestCase xmlns="https://www.steepgraph.com"
7      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
8      xsi:schemaLocation="https://www.steepgraph.com ../../src/main/resources/TestAutomationFrameworkDesktop.xsd"
9      targetApplication="desktop">
10
11    <LaunchApplication
12        filepath="C:\Program Files\Dassault Systemes\B421\win_b64\code\bin\CATSTART.exe" />
13
14    <!-- TBD : Step description -->
15@    <ClickElement locatorImage="1586509297049.png" />
16
17    <!-- TBD : Step description -->
18    <SendKeys key="3dx_user2" />
19
20@    <!-- TBD : Step description -->
21    <ClickElement locatorImage="1586509341835.png" />
22
23@    <!-- TBD : Step description -->
24    <SendKeys key="Passport1" />
25
26@    <!-- TBD : Step description -->
27    <ClickElement locatorImage="1586509371118.png" />
28@>
29  </TestCase>

```

13. DragAndDrop:

With the help of this tag, user should be able to drag and drop the elements and scroll bar from source to target position as per inputs provided by user.

- **Syntax:**

*<DragAndDrop sourceRefId="" sourceLocatorImage="" targetRefId=""
targetLocatorImage="" targetOffset="" />*

NOTES:

1. The images captured for locatorImage attribute should always be in .png extension. The image captured for the required element should be focused and captured properly.
2. In DragAndDrop tag, all attributes are optional, but user must choose one of source and target attribute from the list of attributes provided. For more details on same, please refer to Attributes section for this tag.

- **Attributes:**

- **sourceRefId:**

This attribute is optional. If not provided then sourceLocatorImage will be considered. This refers to the Id, which user needs to get from FindElement Tag and also this refers to source position/element on which drag operation would be applied.

- **sourceLocatorImage:**

This attribute is optional. If sourceRefId is present then it will be ignored. This refers to the Source element/position Image name, on which drag operation would be applied.

- **targetRefId:**

This attribute is optional. If not provided then targetLocatorImage or targetOffset will be considered. This refers to the Id, which user needs to get from FindElement Tag and also this refers to target position/element on which drop operation would be applied.

- **targetLocatorImage:**

This attribute is optional, If targetRefId is present then targetLocatorImage will

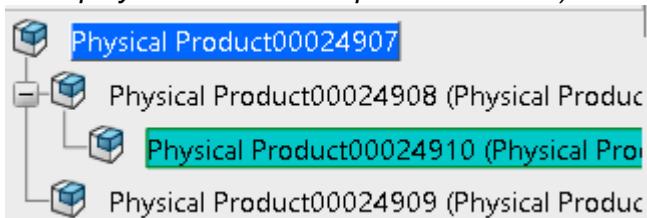
be ignored. This refers to the target element/position Image name, on which drop operation would be applied.

- **targetOffset:**

This attribute is optional, If targetLocatorImage or targetRefId is present then targetOffset will be ignored. This refers to the x and y co-ordinates of target position/element(provided in String format : x,y) on which drop operation would be applied , which x and y is integer values. x and y are offset from the center of the source region (sourceRefId or sourceLocatorImage)

- **Examples:**

Suppose user wants to drag and drop the physical product from one child physical product to another child physical product (i.e. if user wants to drag "Physical Product00024910" from "Physical Product00024908" to "Physical Product00024909" as displayed in screenshot provided below:)

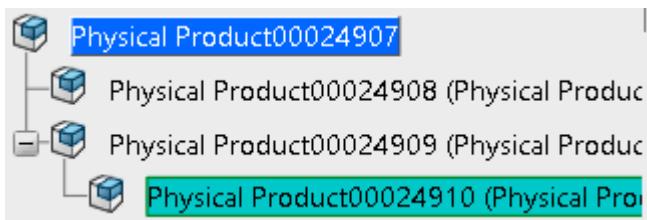


The syntax for the same is provided below in Syntax section:

Syntax:

```
<DragAndDrop sourceLocatorImage="1589549120418.png"
targetLocatorImage="1589549122230.png" />
```

where sourceLocatorImage name is of "Physical Product00024910" and targetLocatorImage name is of "Physical Product00024909". The screenshot of Part structure after completing DragAndDrop operation is provided below:



Created with the Personal Edition of HelpNDoc: [Free EPub producer](#)

ARAS

.Aras Library (Includes Aras tags only)

Created with the Personal Edition of HelpNDoc: [Make CHM Help File Creation a Breeze with HelpNDoc](#)

CheckBox

CheckBox:

This tag is used to click on the check Box.

- **Syntax:**

```
<CheckBox selectCheckBox="Dependent"/>
```

■ **Attribute:**

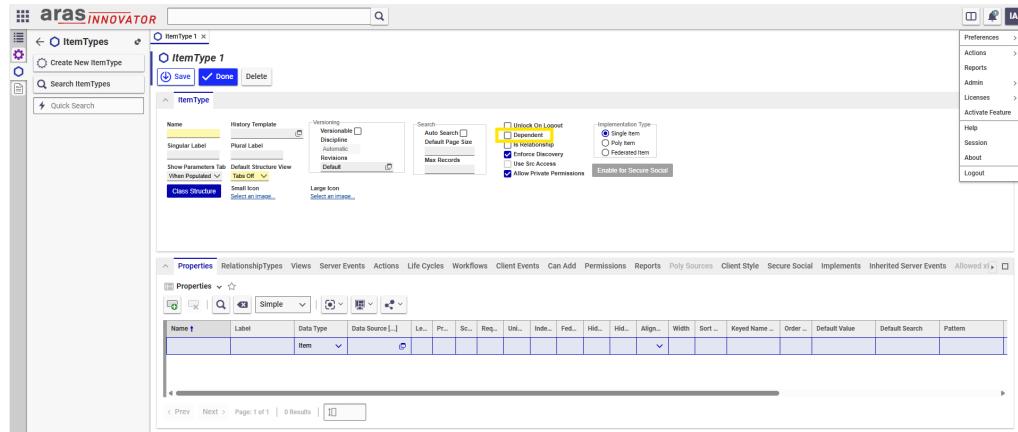
selectCheckBox:(mandatory)

This attribute is mandatory and take input is text of the checkBox.

At a time only one check box can be selected.

■ **Example:**

<CheckBox selectCheckBox="Dependent"/>



Created with the Personal Edition of HelpNDoc: [Converting Word Docs to eBooks Made Easy with HelpNDoc](#)

ClickItemViewCommandBar

ClickItemViewCommandbar:

This tag is used to click the Buttons on ClickItemViewCommandbar of Form. User needs to pass the label of the button, if label is not present pass the name of the tooltip. This tag will also handle the drop down.

■ **Syntax:**

<ClickItemViewCommandbar commandLabel="More|Permissions|Access Report"/>

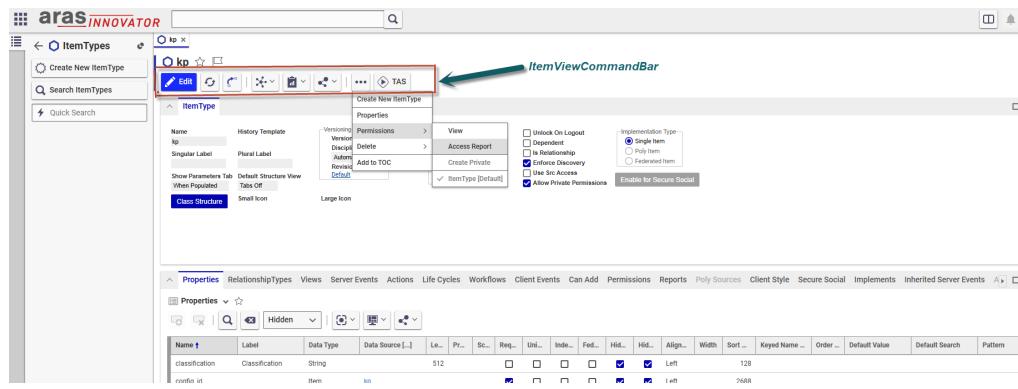
■ **Attribute:**

commandLabel:(mandatory)

The commandLabel attribute takes name of the button or tooltip of button(if label is not present) or in case of drop down if command to execute is inside another command, then user need to pass parent menu/command label and child command label separated by pipe (|).

■ **Example:**

commandLabel = 'More|Permissions|Access Report'



Created with the Personal Edition of HelpNDoc: [Elevate Your Help Documentation with a Help Authoring Tool](#)

ClickRelationshipTab

ClickRelationshipTab:

User can use ClickRelationshipTab to click buttons on relationship bar.

- **Syntax:**

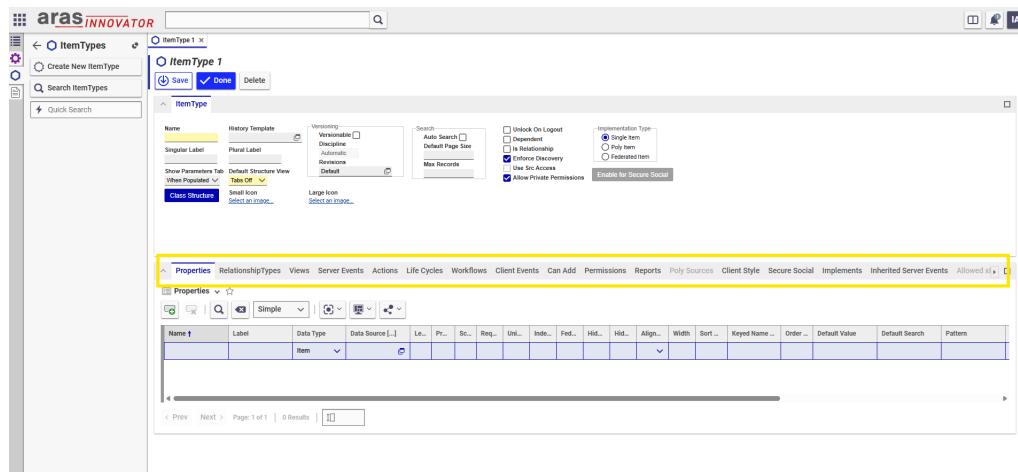
<ClickRelationshipTab commandLabel="Default View Cards"/>

- **Attribute:**

commandLabel:(mandatory)

The commandLabel attribute takes name of the Relationship button

- Please check below image. Suppose user wants to click on a particular command label as in below image, then he can use this tag like
<ClickRelationshipTab commandLabel="Default View Cards"/>



Created with the Personal Edition of HelpNDoc: [Full-featured Documentation generator](#)

CloseNavigationPanel

CloseNavigationPanel:

This tag is used to close the navigationPanel. It will close navigationPanel even if the Panel is pinned or unpinned.

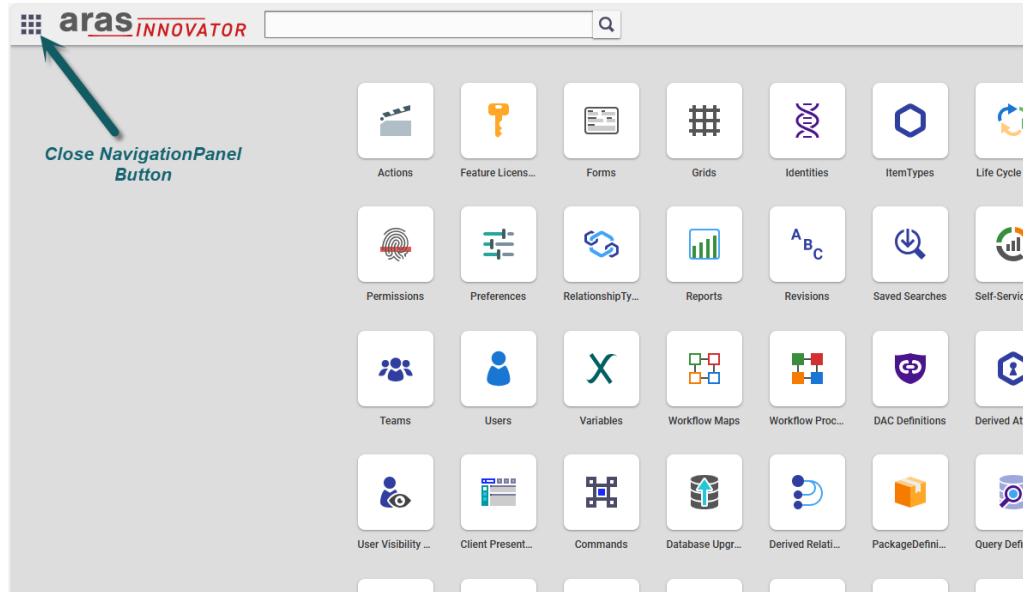
- **Syntax:**

<CloseNavigationPanel/>

- **Attribute:**
EmptyTag

- **Example:**

In the image below, button to open/close Navigation Button is shown.



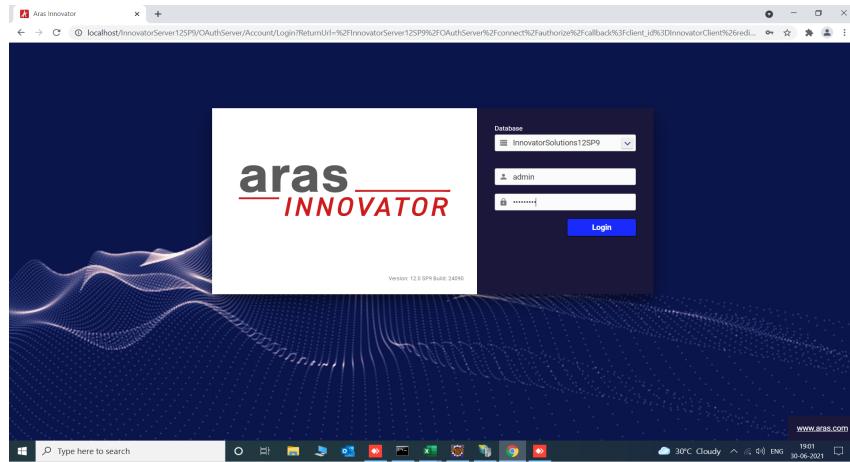
Created with the Personal Edition of HelpNDoc: [5 Reasons Why a Help Authoring Tool is Better than Microsoft Word for Documentation](#)

LogIn

Login:

Login to Aras

- **Syntax:**
`<Login username="" password="" />`
- **Attributes:**
 - **username** : (mandatory) : this attribute provides username for login into Aras.
 - **password** : (mandatory) : this attribute provides password for login into Aras.
- **Examples:**
Please check below image. If user wants to login into Aras innovator, then he can use this tag like
`<Login username="$csv{username}" password="$csv{password}" />`



Created with the Personal Edition of HelpNDoc: [Write eBooks for the Kindle](#)

LogOut

Logout:

Sign out the current user session of Aras.

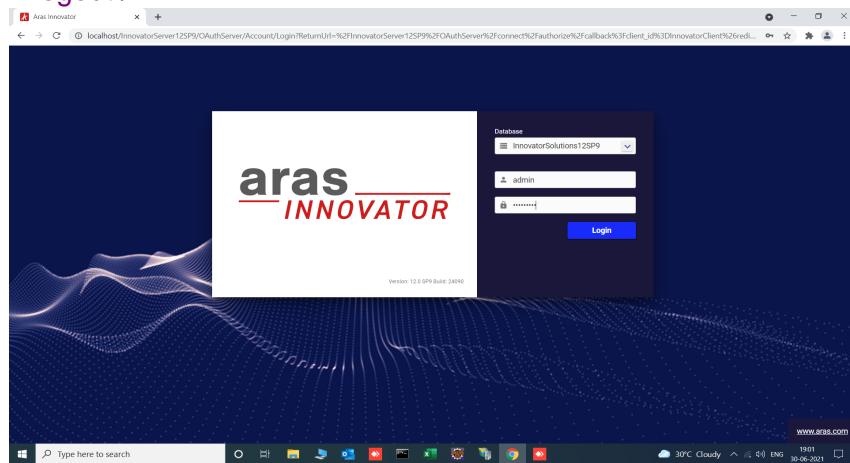
- **Syntax:**

`<Logout />`

- **Examples:**

Please check below image. If user wants to logout of Aras innovator, then he can use this tag like

`<Logout />`



Created with the Personal Edition of HelpNDoc: [Keep Your Sensitive PDFs Safe with These Easy Security Measures](#)

OpenNavigationPanel

OpenNavigationPanel:

This tag is used to open the navigationPanel. It will open and pin the navigationPanel. If the Panel is already open it will just pin it.

■ **Syntax:**

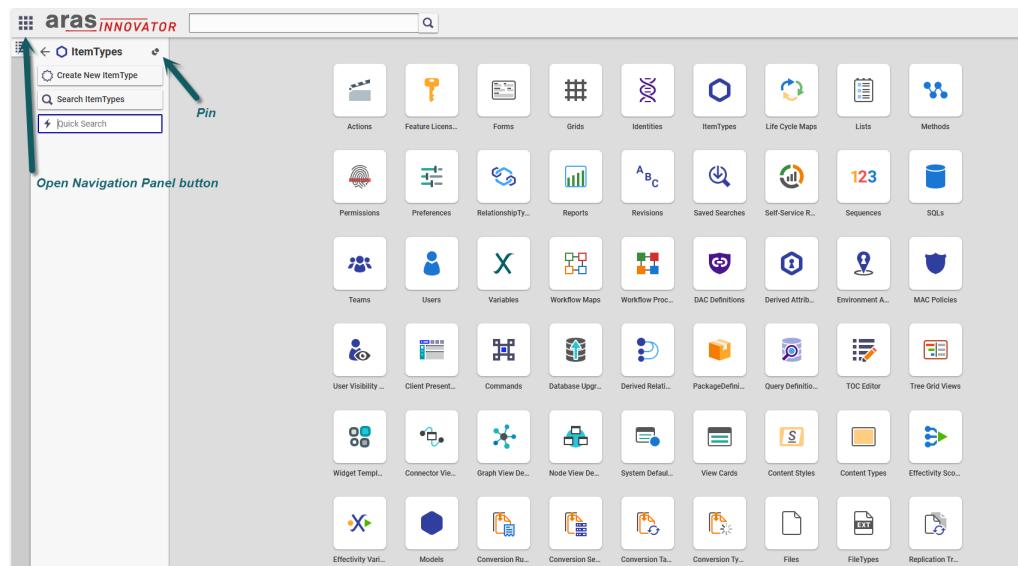
```
<OpenNavigationPanel/>
```

■ **Attribute:**

EmptyTag

■ **Example:**

In the image below, button to open/close Navigation Button is shown along with pin/unpin.



Created with the Personal Edition of HelpNDoc: [Full-featured EBook editor](#)

ShowContents

ShowContents:

User can use ShowContents to click on options inside multi level sub menus of main menu in Navigation panel.

■ **Syntax:**

```
<ShowContents commandLabel="Administration/Configuration/Graph  
Navigation/Graph View Definitions/Create New Graph View Definition"  
highLightElement="true"/>
```

■ **Attribute:**

commandLabel:(mandatory)

The commandLabel attribute takes name of the options in main menu or options in sub menu from main menu, if command to execute is inside another command, then user need to pass parent menu/command label and child command label separated by pipe (|). At the end of this user has to pass keywords "Create" or "Search". Create will open the Item Form, and Search will open the searchGrid.

highlight:(optional)

This attribute used is choose whether to highlight the element when working on it. Takes values true or false. Default value is depends on SG_TAS

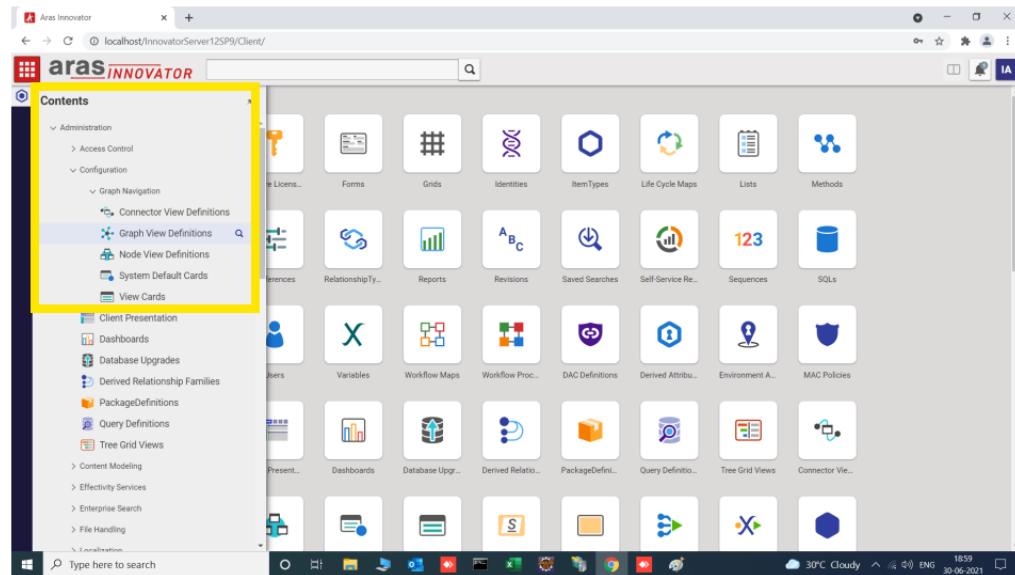
PROPERTIES KEY - sg-tas.highlight.webelement.enable OR fasle if key not defined.

This Attribute will highlight the element for which you have given XPath, CSSSelector,id or name. As shown in Below

style:(optional)

This attribute is used to provide styling to the highlight used. (This can only work when highlight="true" else it will ignore the style attribute.)

- Please check below image. Suppose user wants to click on a particular command label as in below image, then he can use this tag like
`<ShowContents commandLabel="Administration/Configuration/Graph Navigation/Graph View Definitions/Create New Graph View Definition" highlightElement="true"/>`



Created with the Personal Edition of HelpNDoc: Effortlessly Edit and Export Markdown Documents

SwitchToFrameByTitle

SwitchToFrameByTitle:

Using this Tag user can directly switch between frames based on the given title.

- **Syntax:**

```
<switchToFrameByTitle title="Form|Graph View Definition"/>
```

- **Attribute:**

title:(mandatory)

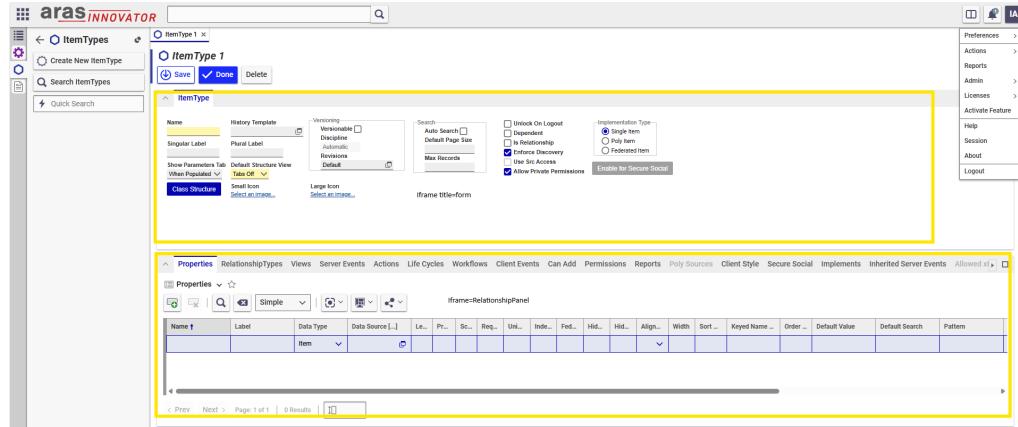
The attribute takes the input as the title of an Iframe. .if user need to pass multiple Iframe title then title will be separated by pipe (|).

- **Note :**

There are five predefined iframes: **From**, **RelationshipPanel**, **DialogPanel**, **SearchGrid**, and **SwitchTab**. If a user wants to navigate inside any of these predefined iframes, they should specify the _____ iframe name directly. However, if the user wants to access elements beyond these predefined iframes (inside them), they should use the format **Form|Graph View Definition**, separating _____ each level with a pipe (|) symbol.

For other Iframes use SwitchToFrame tag.

<switchToFrameByTitle title="Form/Graph View Definition"/>



Created with the Personal Edition of HelpNDoc: Easily create HTML Help documents

UserMenu

UserMenu:

User can use UserMenu to click buttons in Usermenu

- **Syntax:**

<UserMenu commandLabel="Licenses/View Feature Tree"/>

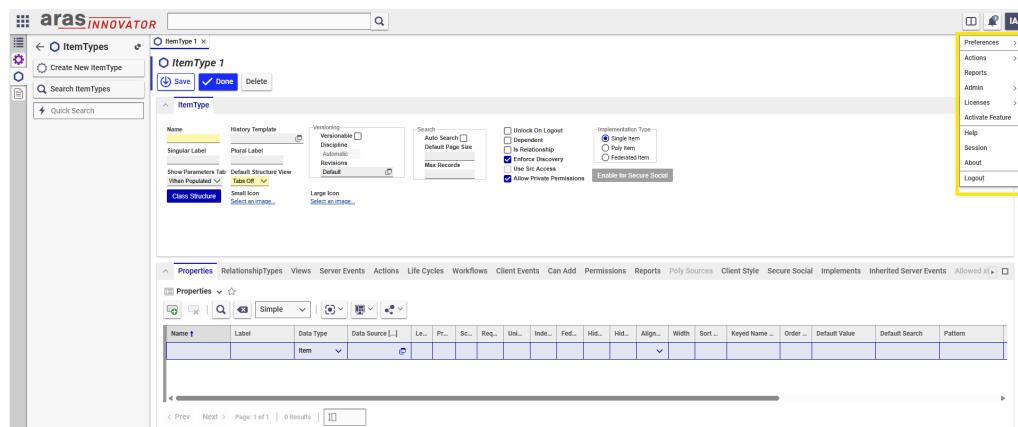
- **Attribute:**

commandLabel:(mandatory)

The commandLabel attribute takes name of the options in main menu or options in sub menu from main menu, if command to execute is inside another command, then user need to pass parent menu/command label and child command label separated by pipe (|).

Please check below image. Suppose user wants to click on a particular command label as in below image, then he can use this tag like.

<UserMenu commandLabel="Licenses/View Feature Tree"/>



Created with the Personal Edition of HelpNDoc: Easily create EPub books

OpenMainWindowApp

OpenMainWindowApp:

Open an app in main window of aras.

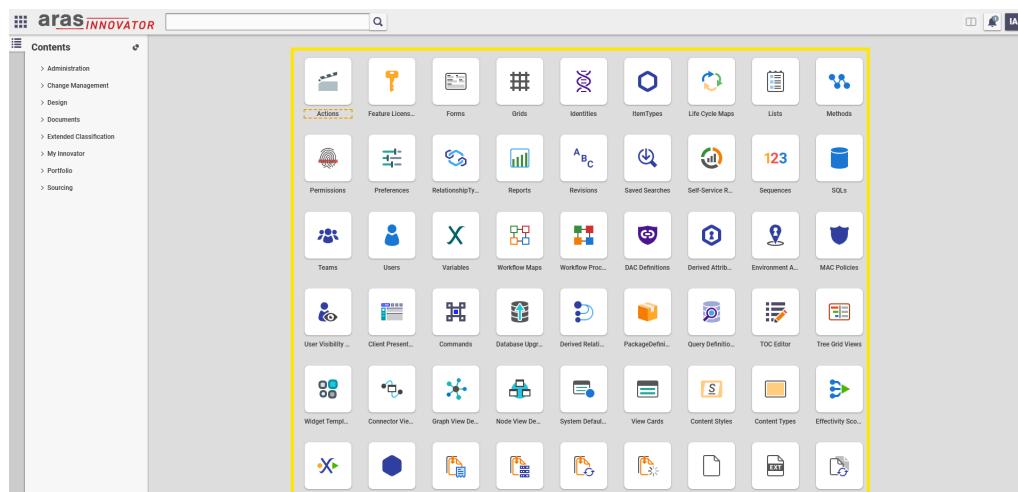
- **Syntax:**

```
<OpenMainWindowApp appName="Actions"/>
```

- **Attribute:**

appName:(mandatory)

The appName attribute takes name of the app to be opened in main Aras window.



Created with the Personal Edition of HelpNDoc: [Free Qt Help documentation generator](#)

Jenkins Installation for SG- TAS-2.1

Created with the Personal Edition of HelpNDoc: [Streamline Your Documentation Process with HelpNDoc's Project Analyzer](#)

1. Introduction

1. *Introduction*

1. [Introduction](#)

2

2. [Software Requirement](#)

2

3. Follow following step if Jenkins are not in your system

2

4. Components to deploy/configure

2

4.1. 3DX-TAS-2.0-WebOnly

2

4.2. Run Jenkins

2

1.1 Introduction

The main purpose of this document is to give assistance for deployment of 3DX-TAS-2.0. This document mainly describes the “How” to Configure and Deploy the 3DX-TAS-2.0 and its components on Jenkins.

Contents

- **What is Jenkins?**

Jenkins is a self-contained, open source automation server which can be used to automate all sorts of tasks related to building, testing, and delivering or deploying software.

Jenkins can be installed through native system packages, Docker, or even run standalone by any machine with a Java Runtime Environment (JRE) installed.

- **About this documentation**

This documentation begins with a Guided Tour to help you get up and running with Jenkins and introduce you to Jenkins’s main feature, Pipeline.

Created with the Personal Edition of HelpNDoc: [Free EBook and documentation generator](#)

2. Software Requirement

2. Software Requirement

- **Jenkins.war**

- **Getting started with the Guided Tour**

This guided tour introduces you to the basics of using Jenkins and its main feature, Jenkins Pipeline. This tour uses the "**standalone**" Jenkins distribution, which runs locally on your own machine.

- **Prerequisites**

For this tour, you will require:

- **A machine with:**

- 256 MB of RAM, although more than 2 GB is recommended
- 10 GB of drive space (for Jenkins)

- **The following software installed:**

- Java 8 or 11 (either a JRE or Java Development Kit (JDK) is fine)
- Download and run **Jenkins**
- Open up a terminal in the download directory.

- Run **java -jar jenkins.war --httpPort=8080**.
- Browse to **http://localhost:8080**.
- Follow the instructions to complete the installation.
- When **the installation is complete, you can start putting Jenkins to work!**

Created with the Personal Edition of HelpNDoc: [Achieve Professional Documentation Results with a Help Authoring Tool](#)

3. Steps

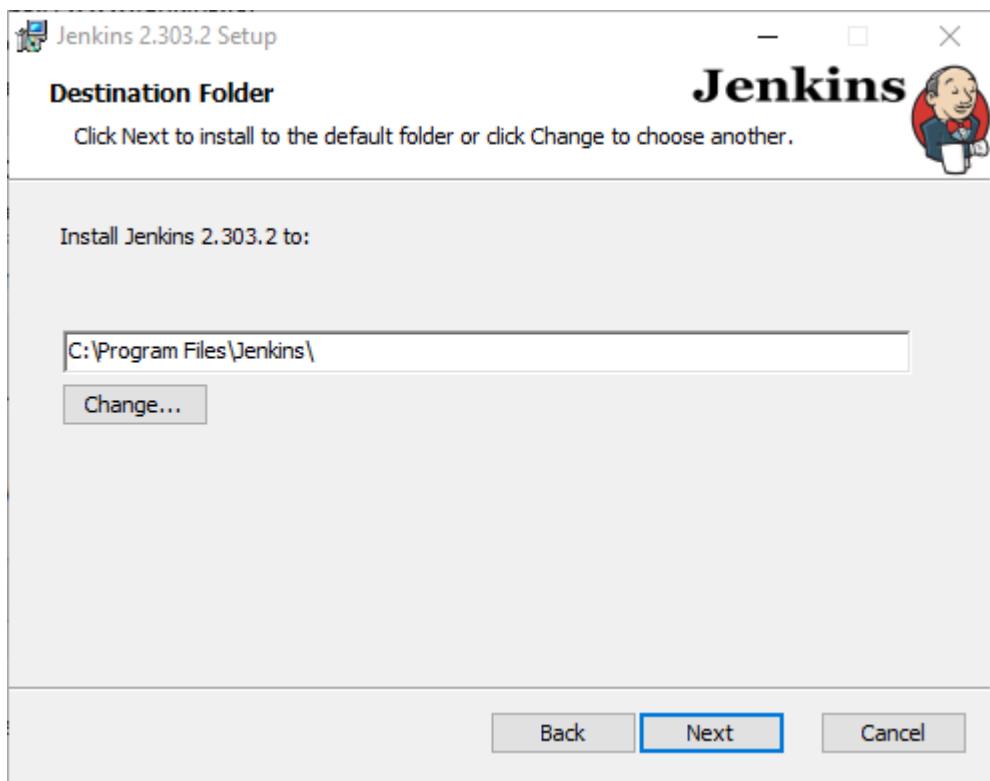
3. There are two ways to installed Jenkins

1. Double click on jenkins application.
2. Using **cmd promt (this one we recommended)**

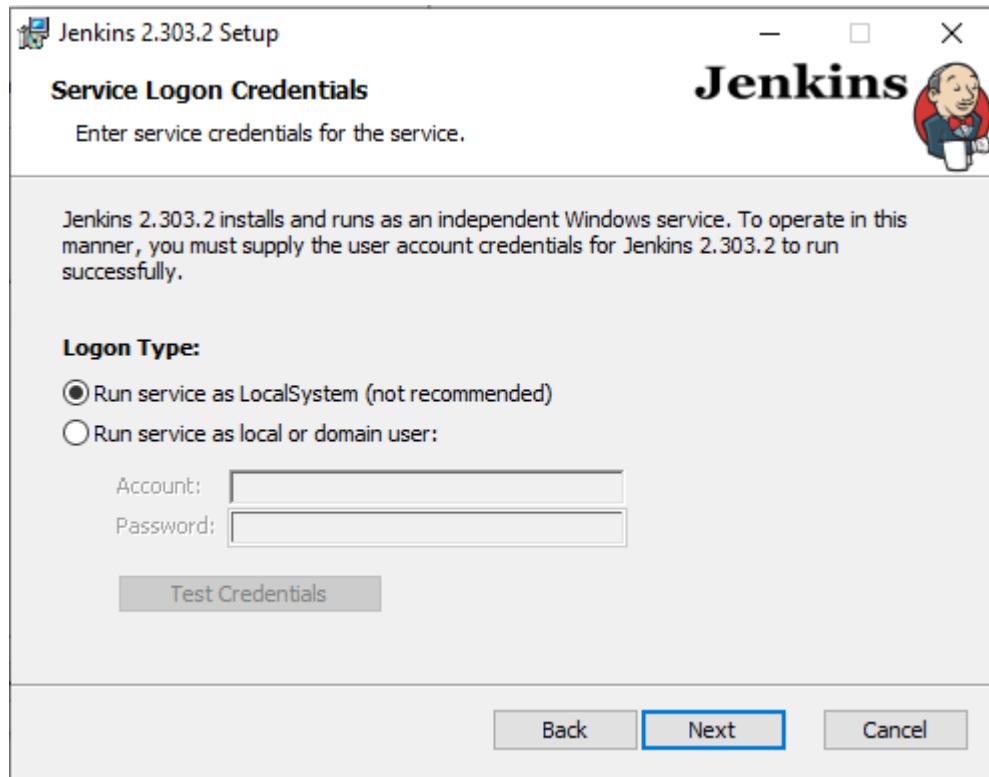
1. Follow following steps for Jenkins Installation Process or if not not in your system

Jenkins is a java application used to continuous integration and continuous delivery.

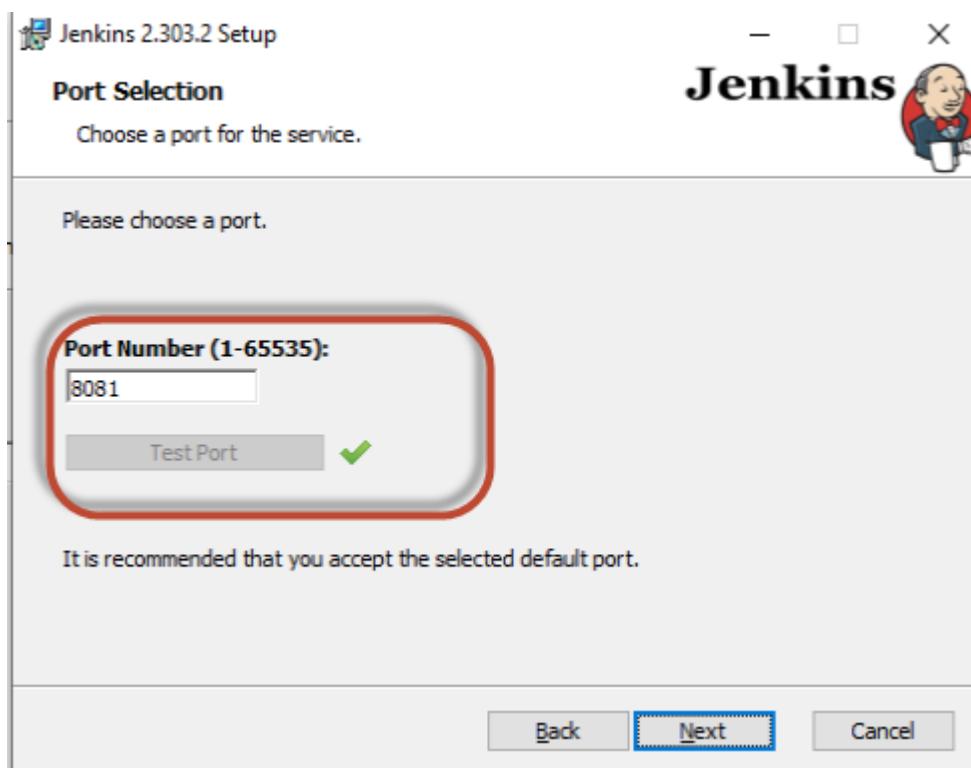
1. Go to url <https://www.jenkins.io/> (for download Stable (LTS) Jenkins.)
 - Place the war file on any location of your system.
2. Open jenkins.war file --> Click on **Next**



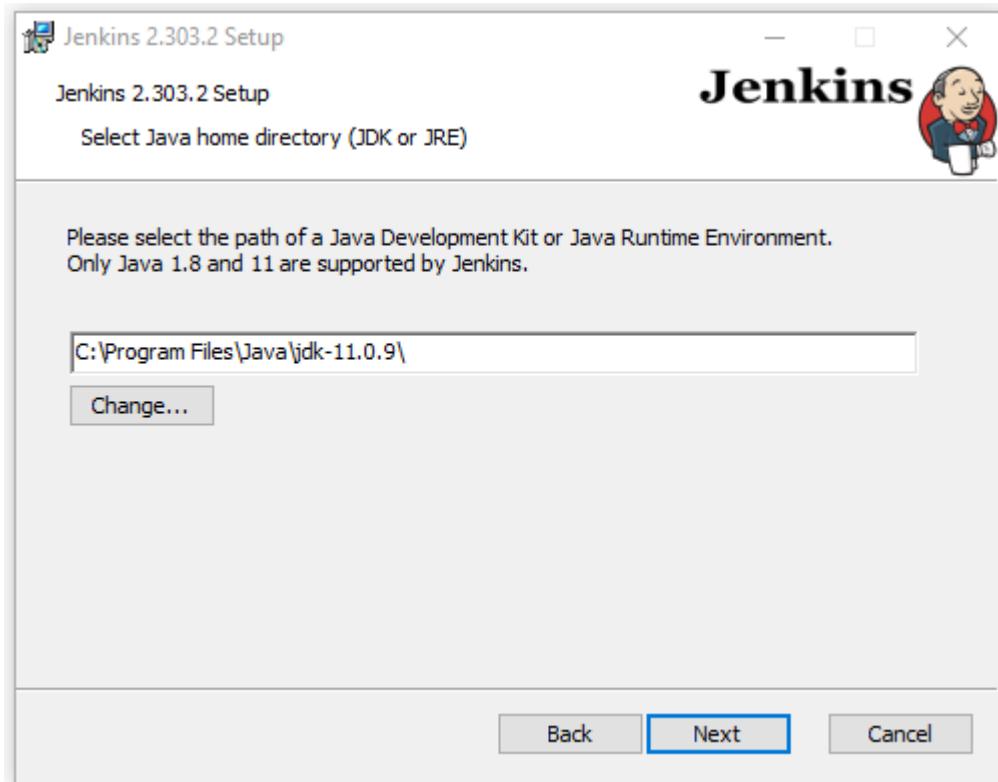
3. Then Choose **Run Service as LocalSystem(not recommended)**



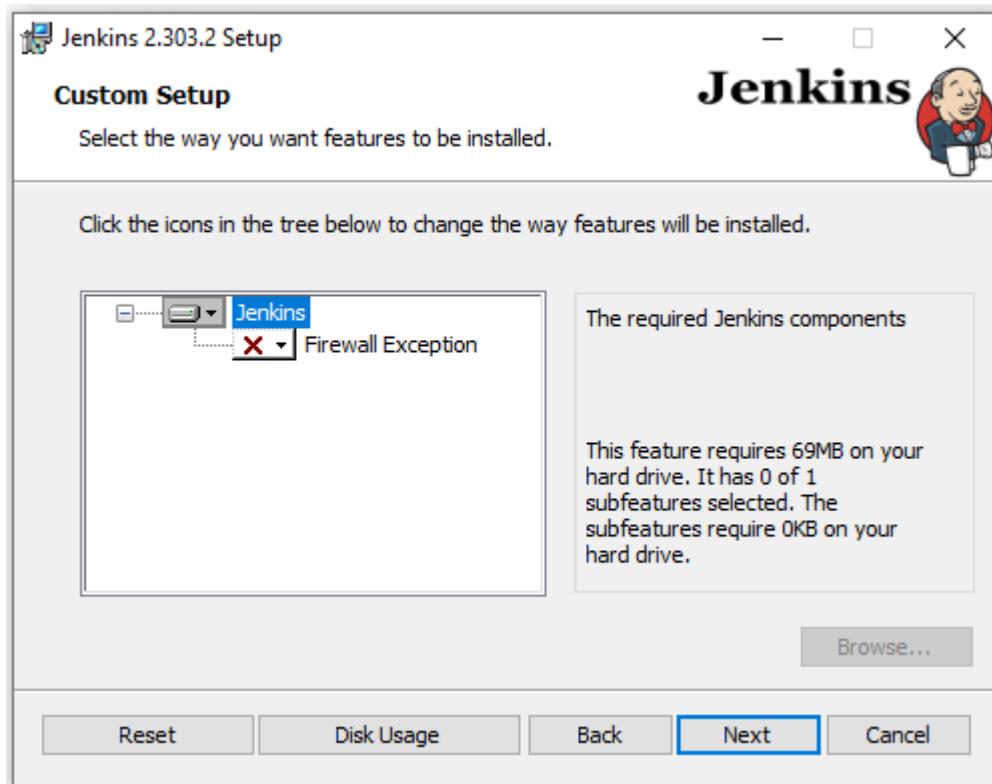
4. Please choose a port or if it's not same port for any other application then live it default port à **Next**



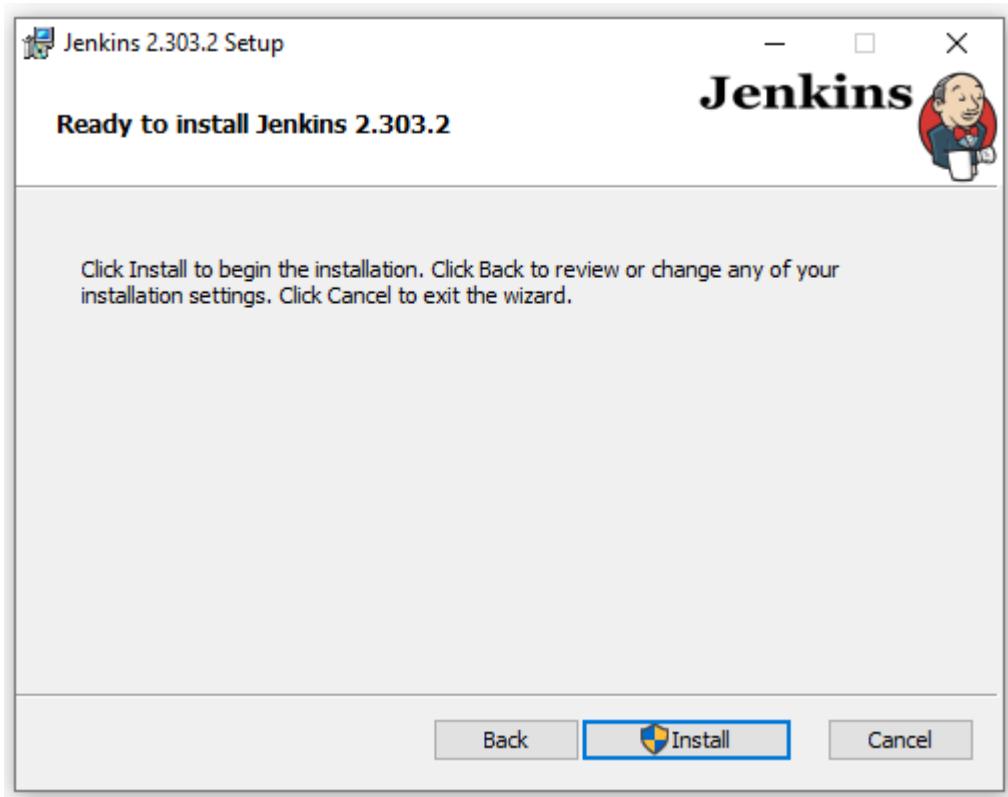
5. Choose **Next**



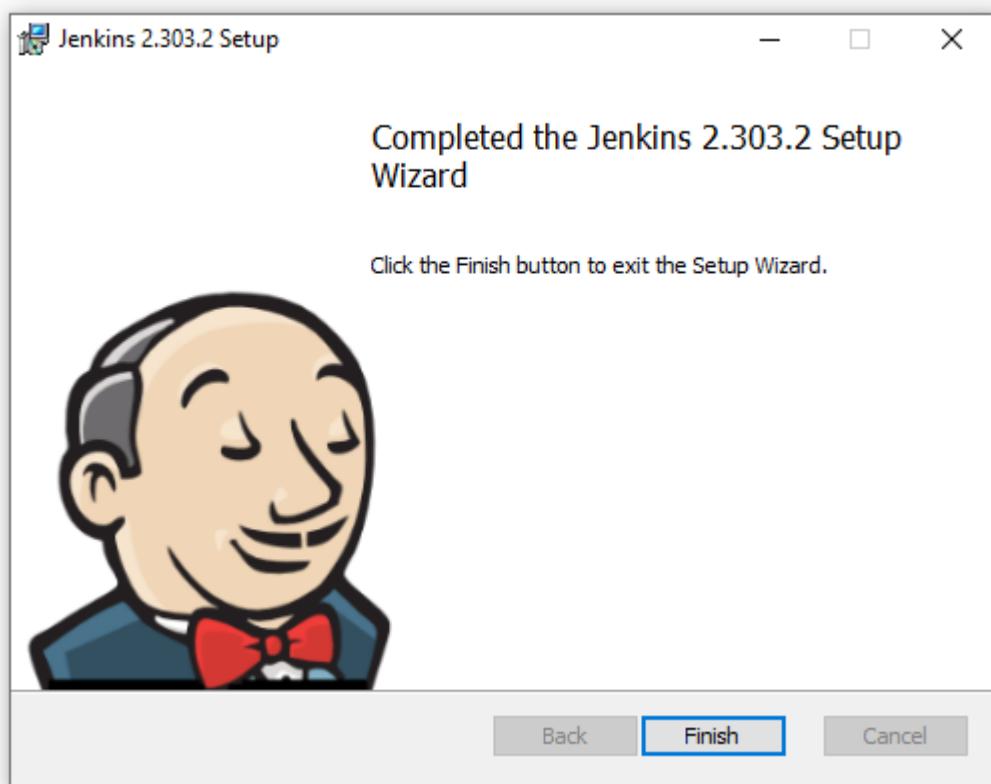
6. Select other services that need to be installed with Jenkins and click on the “Next” button.



7. Click the “Install” button to start the installation process.



8. The installation will proceed.



4.1 3DX-TAS-2.0-WebOnly

3DX-TAS-2.0-WebOnly is rar file. Extract the jar file in the **Jenkins workspace**. Having the project name **3DX-TAS-2.0**.

- Make changes in .properties file.

- Provide the correct driver.
- Provide proper license.
- Place testcase in testsuits.

Created with the Personal Edition of HelpNDoc: Experience a User-Friendly Interface with HelpNDoc's Documentation Tool

4. Components to deploy/configure

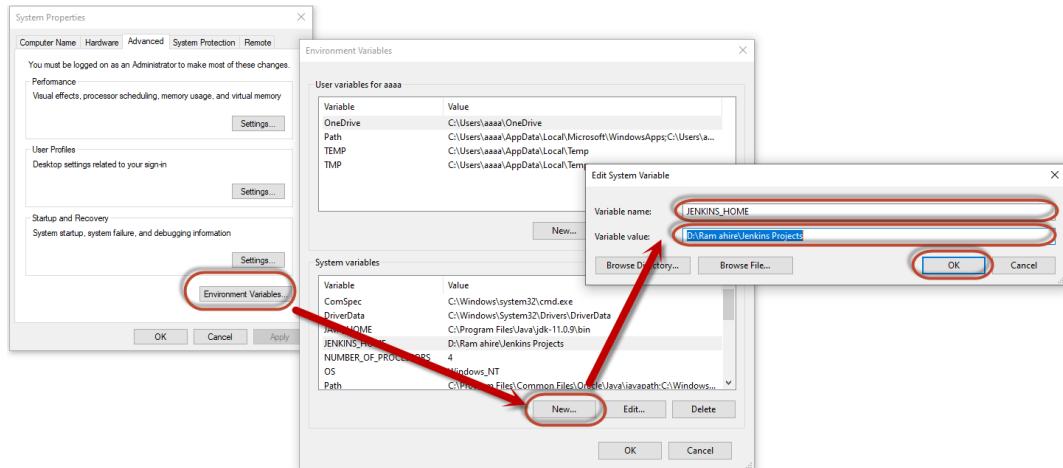
4. Using cmd prompt (this one we recommended)

Components to deploy/configure

There will be 1 component as a part of 3DX-TAS-2.0

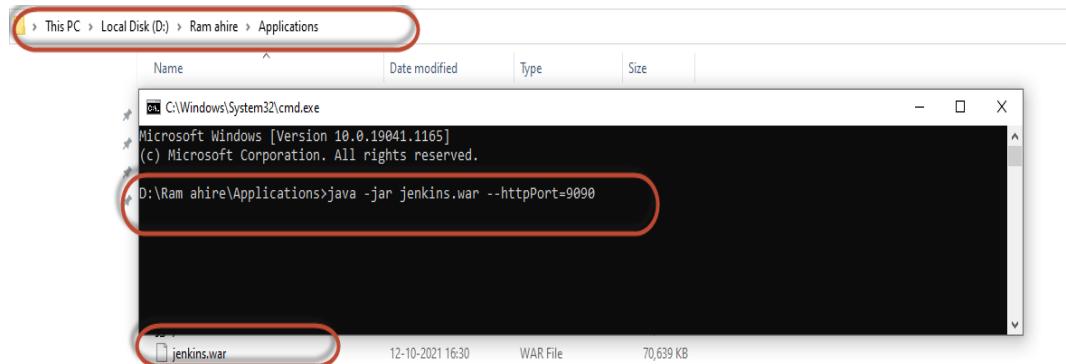
- a. Create Folder Structure and Kept jenkins.war inside the folder like "D:\FolderName\Jenkins Projects" --> jenkins.war
- b. Set the path folder in **Computer Properties** --> **Advance Settings** --> **Environments Variables** --> **System variable** --> Click on **New** --> **JENKINS_HOME** --> **Folder_Path** --> Ok (Like below mention)
- c. **JENKINS_HOME = D:\Ram ahire\Jenkins_Projects --> Jenkins_Workspace**
in environment variable in windows OR in Linux type cmd: **export JENKINS_HOME = /Users/admin/Jenkins/Jenkins_Workspace**

NOTE: jenkins home path does not contain space



- d. Run Jenkins

Open the **Jenkins folder where jenkins.war file has** --> Open CMD prompt --> write "**java -jar jenkins.war --httpPort=9090**" --> Enter



- e. To unlock Jenkins, copy the password from the file `initialAdminPassword`. This file should be found under the Jenkins installation path.

Jenkins initial setup is required. An admin user has been created and a password generated.
Please use the following password to proceed to installation:
22f2d3b6c493480683014ffc977f0519

- f. Copy the content of the `initialAdminPassword` file and paste it in the Administrator password field. Then, click the "Continue" button.

Getting Started

Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to the log ([not sure where to find it?](#)) and this file on the server:

D:\Ram ahire\Jenkins Projects\secrets\initialAdminPassword

Please copy the password from either location and paste it below.

Administrator password

.....

Continue

- Wait until the plugins are completely installed.

Getting Started

Getting Started

Folders	OWASP Markup Formatter	Build Timeout	Credentials Binding
Timestamper	Workspace Cleanup	Ant	Gradle
Pipeline	Github Branch Source	Pipeline: GitHub Groovy Libraries	Pipeline: Stage View
Git	SSH Build Agents	Matrix Authorization Strategy	PAM Authentication
** SSH server Folders ** Trilead API OWASP Markup Formatter ** Structs ** Pipeline: Step API ** Token Macro			
** - required dependency			

- The next thing that you should do is create an Admin user for Jenkins. Then, enter your details and click Save and Continue.

Getting Started

NewItem623 (Source: Includ

Create First Admin User

Username: ramahire

Password: *****

Confirm password: *****

Full name: Ram Ahire

E-mail address: rama@steepgraph.com

Jenkins 2.303.2 Skip and continue as admin **Save and Continue**

- Click on Save and Finish to complete the Jenkins installation.

Getting Started

Instance Configuration

Jenkins URL:

The Jenkins URL is used to provide the root URL for absolute links to various Jenkins resources. That means this value is required for proper operation of many Jenkins features including email notifications, PR status updates, and the `BUILD_URL` environment variable provided to build steps.

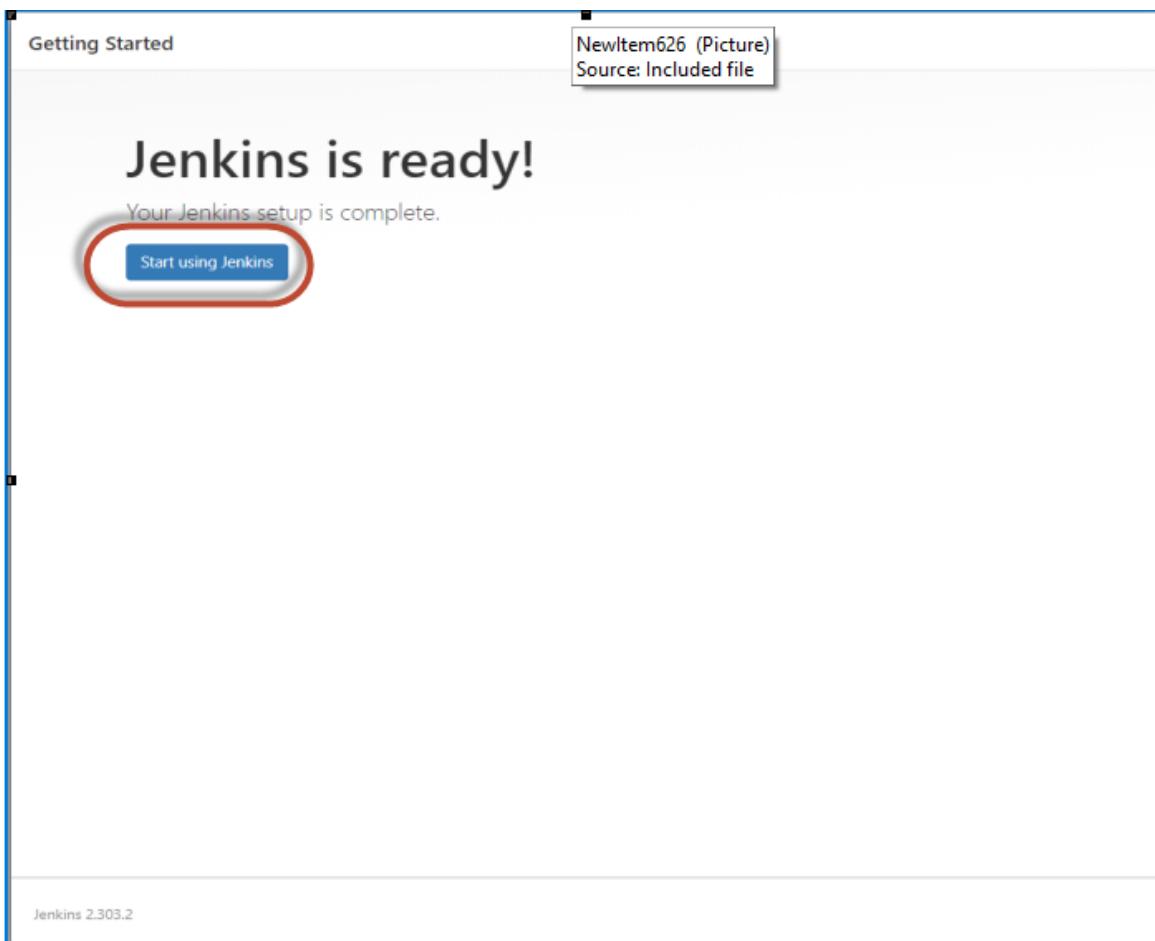
The proposed default value shown is **not saved yet** and is generated from the current request, if possible. The best practice is to set this value to the URL that users are expected to use. This will avoid confusion when sharing or viewing links.

Jenkins 2.303.2

Not now

Save and Finish

- Now, click on Start using Jenkins to start Jenkins.



4.1 Run Jenkins

Follow the following steps to run Jenkins.

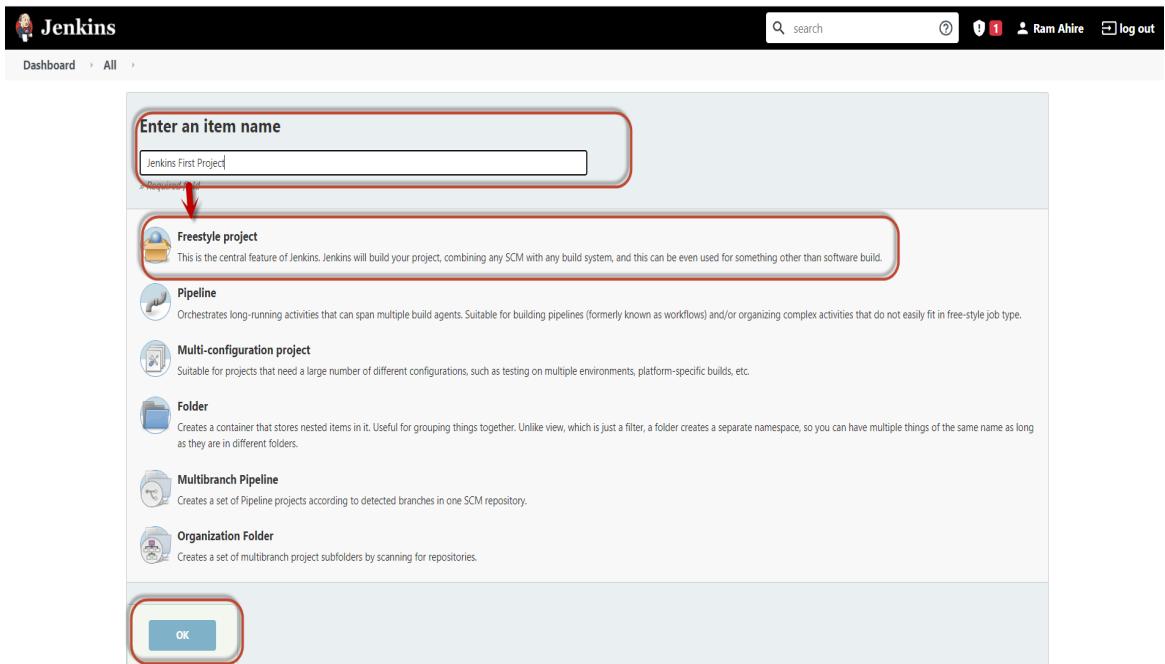
- Go to the <http://localhost:9090/>
- **Finally, here is the default Jenkins page.**

The screenshot shows the Jenkins dashboard at localhost:8081. The main content area displays the "Welcome to Jenkins!" message with instructions to start building a software project. It includes sections for "Create a job", "Set up a distributed build", and "Build Executor Status". The sidebar on the left contains links like "Dashboard", "New Item", "People", "Build History", "Manage Jenkins", "My Views", "Lockable Resources", and "New View". The "New Item" link is circled in red. The bottom right corner shows "REST API" and "Jenkins 2.303.2".

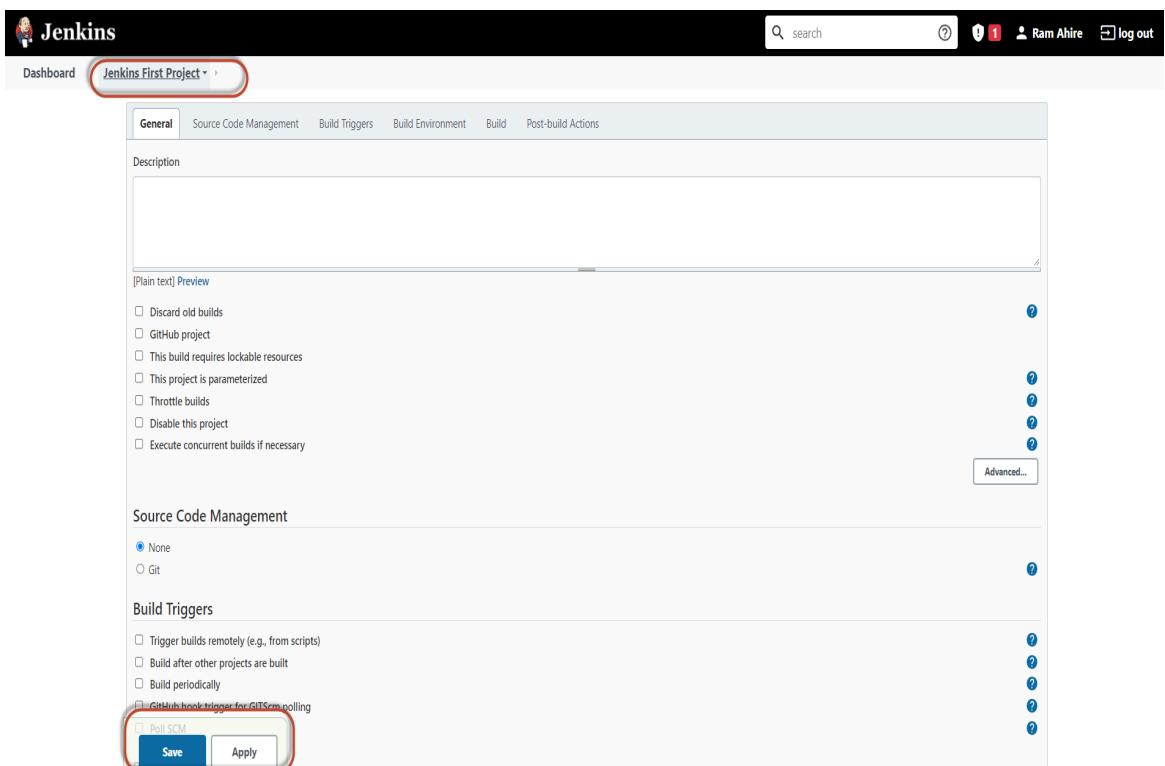
- Click on **New Item** for creating Project

This screenshot is identical to the one above, showing the Jenkins dashboard at localhost:8081. The "New Item" button in the sidebar is explicitly circled in red. The rest of the interface, including the central "Welcome to Jenkins!" message and the sidebar links, remains the same.

- Create a job by Click **New Item** à Enter Project Name i.e **3DX-TAS-2.0** à Click on **Freestyle Project** à Click **Ok**.



- After clicking **OK** button then appear new window same as project name created --> Click on **Apply and Save**



- Create of Project name **3DX-TAS-2.0** is created.

The screenshot shows the Jenkins dashboard. On the left, there's a sidebar with links like 'New Item', 'People', 'Build History', 'Manage Jenkins', 'My Views', 'Lockable Resources', and 'New View'. Below that are two expandable sections: 'Build Queue' (which says 'No builds in the queue.') and 'Build Executor Status' (which shows 1 idle and 2 idle). The main area is titled 'All' and lists a single build: '3DX-TAS-2.0' with a green checkmark icon. It shows 'Last Success' at '2 hr 47 min - #6', 'Last Failure' at '3 hr 19 min - #1', and 'Last Duration' at '26 sec'. There are also links for 'Atom feed for all', 'Atom feed for failures', and 'Atom feed for just latest builds'.

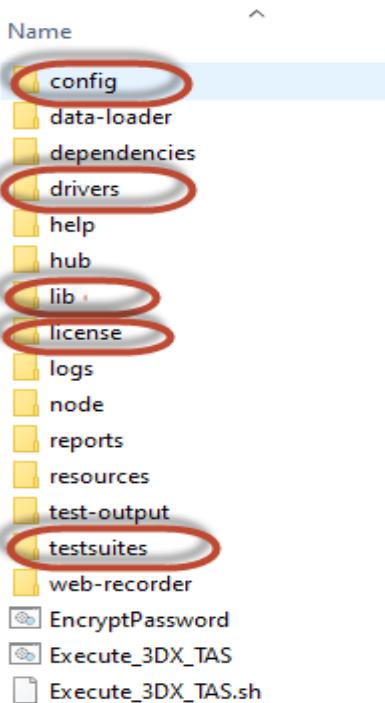
After creating Project Go to project folder then you see the workspace folder , Double click on that folder -> You see Project Name which you created through Jenkins which shown below

This PC > New Volume (D:) > Jenkins_Projects_Workspace				
Name	Date modified	Type	Size	
jobs	20-10-2021 02:06 PM	File folder		
logs	20-10-2021 12:42 PM	File folder		
nodes	20-10-2021 12:42 PM	File folder		
plugins	20-10-2021 12:53 PM	File folder		
secrets	20-10-2021 02:46 PM	File folder		
updates	20-10-2021 12:53 PM	File folder		
userContent	20-10-2021 12:42 PM	File folder		
users	20-10-2021 12:57 PM	File folder		
war	20-10-2021 12:42 PM	File folder		
workflow-libs	20-10-2021 12:53 PM	File folder		
workspace	20-10-2021 02:13 PM	File folder		
.laststarted	21-10-2021 10:28 AM	LASTSTARTED File	0 KB	
.owner	20-10-2021 11:46 PM	OWNER File	1 KB	
config	21-10-2021 10:28 AM	XML Document	2 KB	
hudson.model.UpdateCenter	21-10-2021 10:28 AM	XML Document	1 KB	
hudson.plugins.git.GitTool	20-10-2021 12:53 PM	XML Document	1 KB	
identity.key.enc	20-10-2021 12:42 PM	ENC File	2 KB	
jenkins.install.InstallUtil.lastExecVersion	21-10-2021 10:28 AM	LASTEXECVERSIO...	1 KB	
jenkins.install.UpgradeWizard.state	20-10-2021 12:57 PM	STATE File	1 KB	
jenkins.model.JenkinsLocationConfigura...	20-10-2021 12:57 PM	XML Document	1 KB	
jenkins.telemetry.Collector	20-10-2021 12:42 PM	XML Document	1 KB	
nodeMonitors	21-10-2021 10:28 AM	XML Document	1 KB	
queue	21-10-2021 10:29 AM	XML Document	1 KB	
queue.xml.bak	20-10-2021 06:24 PM	BAK File	1 KB	
secret.key	20-10-2021 12:42 PM	KEY File	1 KB	
secret.key.not-so-secret	20-10-2021 12:42 PM	NOT-SO-SECRET F...	0 KB	

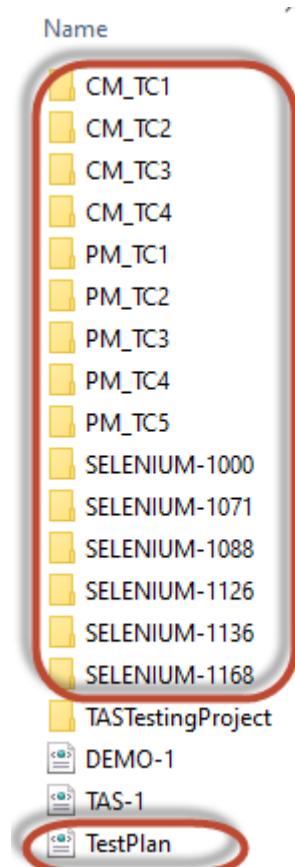
This PC > New Volume (D:) > Jenkins_Projects_Workspace > workspace				
Name	Date modified	Type	Size	
3DX-TAS-2.0	20-10-2021 02:59 PM	File folder		
First_jenkins_projects	20-10-2021 05:37 PM	File folder		

- Double Click on **Specific Project** and **Change Config** file which is required, Update

current driver, lib , and Update working license file like shown below



- Go to testsuites folder and give past testcases which you want to run through jenkins and update TestPlan.xml file. like shown below



- Go to Jenikins Dashboard, Click on Project 3DX-TAS-2.0 à Click on Configure

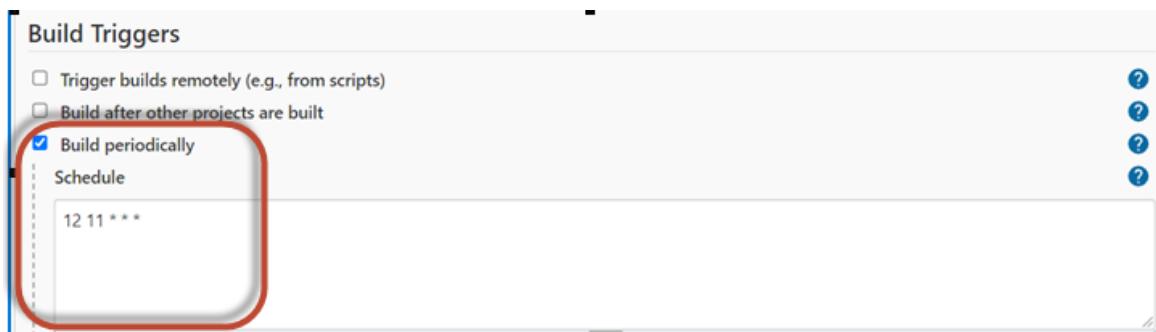
Project 3DX-TAS-2.0

- Workspace
- Recent Changes

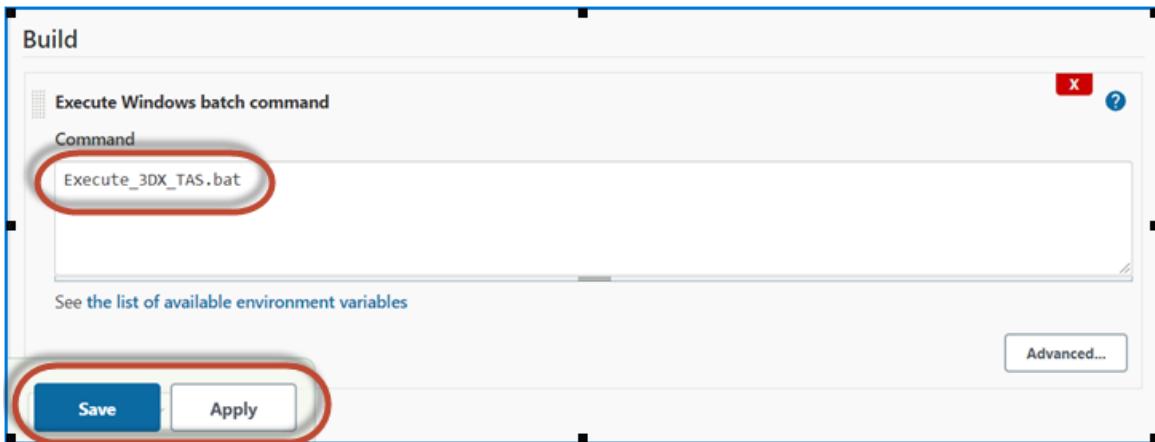
Permalinks

- Last build (#6), 2 hr 49 min ago
- Last stable build (#6), 2 hr 49 min ago
- Last successful build (#6), 2 hr 49 min ago
- Last failed build (#1), 3 hr 21 min ago
- Last unsuccessful build (#1), 3 hr 21 min ago
- Last completed build (#6), 2 hr 49 min ago

- Maintain Schedule time : In **Build Triggers** section select **Build periodically**.
- MINUTES Minutes in one hour (0-59)
 - HOURS Hours in one day (0-23)
 - DAYMONTH Day in a month (1-31)
 - MONTH Month in a year (1-12)
 - DAYWEEK Day of the week (0-7) where 0 and 7 are Sunday



- In build section provide the name of .bat or .sh to execute.

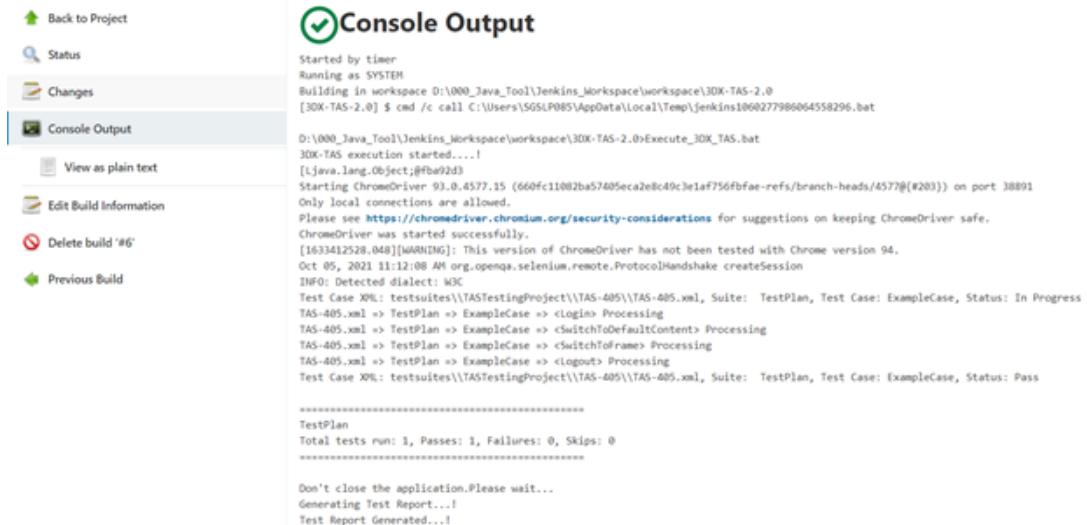


- Click **Apply** à click **Save**.
- **Click on**
- Build will get execute according to the periodic time.
- Go to the Build History section for console and output.

The screenshot shows the 'Build History' page. At the top, there is a search bar containing the text 'find'. Below the search bar is a list of six build entries, each consisting of a checkmark icon, a circled number, and a timestamp. The entry '#6' is highlighted with a red circle around its checkmark and number. The other entries are: #5 (Oct 5, 2021 11:11 AM), #4 (Oct 5, 2021 10:55 AM), #3 (Oct 5, 2021 10:48 AM), #2 (Oct 5, 2021 10:43 AM), and #1 (Oct 5, 2021 10:40 AM).

#	Date
6	Oct 5, 2021 11:12 AM
5	Oct 5, 2021 11:11 AM
4	Oct 5, 2021 10:55 AM
3	Oct 5, 2021 10:48 AM
2	Oct 5, 2021 10:43 AM
1	Oct 5, 2021 10:40 AM

- Click **on # href link to open**



The screenshot shows the Jenkins interface with the 'Console Output' tab selected. The output window displays the log of a build named '30X-TAS-2.0'. The log shows the following:

```

Started by timer
Running as SYSTEM
Building in workspace D:\000_Java_Tool\Jenkins_Worksplace\workspace\30X-TAS-2.0
[30X-TAS-2.0] $ cmd /c call C:\Users\SGSLP085\AppData\Local\Temp\jenkins1060277986064558296.bat
D:\000_Java_Tool\Jenkins_Workspace\workspace\30X-TAS-2.0>Execute_30X_TAS.bat
30X-TAS execution started...
[Ljava.lang.Object;@fbfa92d3
Starting ChromeDriver 93.0.4577.15 (660fc11002ba57405eca2e8c49c3e1af756fbfae-refs/branch-heads/4577[#203]) on port 38891
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.
ChromeDriver was started successfully.
[1633412528.048][WARNING]: This version of ChromeDriver has not been tested with Chrome version 94.
Oct 05, 2021 11:12:08 AM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: W3C
Test Case XML: testsuites\\TASTestingProject\\TAS-405\\TAS-405.xml, Suite: TestPlan, Test Case: ExampleCase, Status: In Progress
TAS-405.xml => TestPlan => ExampleCase => <Login> Processing
TAS-405.xml => TestPlan => ExampleCase => <SwitchToDefaultContent> Processing
TAS-405.xml => TestPlan => ExampleCase => <SwitchToFrame> Processing
TAS-405.xml => TestPlan => ExampleCase => <Logout> Processing
Test Case XML: testsuites\\TASTestingProject\\TAS-405\\TAS-405.xml, Suite: TestPlan, Test Case: ExampleCase, Status: Pass
=====
TestPlan
Total tests run: 1, Passes: 1, Failures: 0, Skips: 0
=====

Don't close the application. Please wait...
Generating Test Report...
Test Report Generated...

```

- In **Console Output** we can see the **Tags execute successfully.**

Created with the Personal Edition of HelpNDoc: [Full-featured Kindle eBooks generator](#)