Integration with Ranorex

() OW mate you'll learn

- Ranorex concepts and mapping to Xray
- Prerequisiteearn Ranorex core concepts
- Implement Legan howated atings test automation test cases looks like, at high-level, using Ranorex
- Running thStudios
 - RRuningeteestsuandprushcthextestoreport to Xray
 - RValidage that the results and available in Jira
- IntegratingLearnthey to assess the impacts on related user stories in Jira, using Xray
- Tips
 - · Seeing the impacts of test automation results on user stories or requirements
 - Run iterations and data-driven tests
 - Run iterations
 - Data-driven tests
 - ° Ranorex's built-in integration with Jira

• References Overview

Ranorex can be seen as a keyword-driven framework for implementing GUI test automation across a bro ad set of technologies, including desktop, web, and mobile. It's a codeless test automation solutiion. Ranorex provides comprehensive support for test automation as detailed in its core features. Data-driven testing is also supported.

Ranorex provides Ranorex Studio, the main application to implement and organize automated test scripts. Ranorex Studio has multiple components, including a recorder (Ranorex Recorder) and a object /element identifier (Ranorex Spy).

In this article we'll highlight some of Ranorex core concepts and see how to have visilbity of test automation results in Jira, using Xray.

Integrating with Xray is straightforward, using JUnit XML reports that Ranorex can generate. To get the integration done, you just need to get that working.

Ranorex concepts and mapping to Xray

Ranorex Studio provides a complete GUI for implementing automated tests. Therefore, we find some concepts typical in IDEs (e.g. solution, project).

There are some specifics though, related to test automation.

The only concept with a direct mapping to Xray will be Ranorex' Test case which will be abstracted as a Xray Test issue (unstructured/generic).

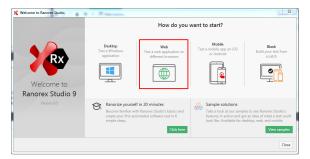
Ranorex concept	Description	Xray concept
Solution	In Ranorex Studio, a solution is the top-level container that contains all other test files. Solutions are organized into one or more projects. Whenever creating a solution, we may identify the type of application we aim to test (e.g. desktop, web, mobile). A solution has always a "test suite" project.	
Project	A tailored place to organize test files. A project can be of one of several types, including "test suite", which offer different capabilities.	
Test suite	The test suite is where you build, organize, and run your tests in Ranorex Studio. A test suite consists primarily of test cases.	Doesn't exist as an entity. Will be visible and part of the definition of each Test issue.
Test case	A test, composed of Modules, which in turn are composed of Actions.	Test issue.

Module	A modularized sequence of actions, that have a certain goal. Can be seen as a grouped sequence of steps in a test case. Modules can be reused between Test cases.	
Action	A step, inside a Module. Can be a mouse/keyboard interaction or a validation.	
Validation	An assertion, a validation.	
Repository	 A Repository contains Repository Items (i.e. UI elements) organized in a tree-like structure. UI elements that contain other UI elements are represented as folders in the repository, with app folders acting as top-level elements and rooted folders as children. 	
Repository item	A representation of a user interface (UI) element used in a test. Each repository item has a name and is defined by its RanoreX Path (i.e. path).	

The overall result of a given test case will be available on the Test Run that will be created in Xray and associated to the corresponding Test issue.

Prerequisites

We start by creating a solution, web-based one as we aim to test a web site.



Under "Additional options" we can customise the language of the underlying code that Ranorex will use to support the test automation.

reate your new solution	5,0)	
e Basic data	Web application Configure behavior	Finish
	Create your new solution	
Solution name	MyTest1	
Location	d:\Documents\Ranorex\RanorexStudio Projects\	
Additional options	d:\Documents\Ranorex\RanorexStudio Projects\	
Project name	MyTest1	
Language	C# ~	
	Create directory for solution Add solution to source control	
		«Back Continu

A project of type "test suite" will be created.

It contains one Test Suite, where we can create and manage our test cases. We may rename this Test Suite to have a more meaningful name (e.g. "Authentication").

NyTestt - Resores Studio - (32bit)				_		- 0 ×
fie fdit Dew Project Spild Debug Search	Icoh Hindow Help					
💽 O 🔛 🖉 ADO 📑 🞯 🗛 🛱 🛛	👌 view 🖿 🔣 🖏	🖉 🦁 🕏 RUN 🕨 🔳	🕷 BURLD 📥 🏦 SPY 💃 🕮	Defealt leyout ~ ,O		
	X OperBrowser.resc Lo	put.net Authoritecton.edut 3	c resultanteg			
자 (종) C (원 (홍) 🔛 승제 MyTesti	O ADD V	NUN Aljants	Y 🕥 Maintenaece mode		MANAGE DATA BOURCES	🎾 Stadio
A Set Presences	NOCE IN	Data source			Search (FI)	Q
Beports D accounts	1 Zen		Data binding / Readians	Description		
D Assemblicita ca	🛛 💽 Authenticatio					
Authentication.orbit	8 R N Valid,					
Of Closelinowserzanec	8-R 💽 Inul	d_Login				
Contraction of the second seco	18 16 F AND					
fodule browser	B R Data	Driven_trivalid_Login	NewConnector Rows: 3			
aarch (72)	Q					
S The MyTectl	A					
- Croups						
B Modules						
B 90 Data Diven Invalid Login Attempt	1					
BI Login_with_invalid_oredentials						
BH Login, with valid, credentials	1					
- SH Legest	×					

Implementing automated tests

We will implement some automated tests for a dummy web site, providing an authentication mechanism that we aim to check, namely the login and logout features.

User Name: Password: LOGIN Welcome Page Login succeeded. Now you can <u>logout</u> .	
-	
-	
-	

Our tests, part of a test suite, include these scenarios:

- valid login
- invalid login
- valid logout
 invalid login (data-driven scenario)

To add some Test cases to the Test Suite, we can use "Add" button.

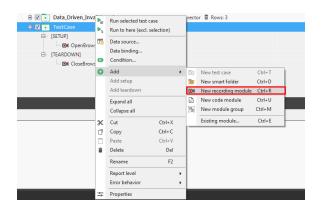
OpenBrowser.rxrec	Logout.rxrec Auther	ntication.rxtst* 🗙	results.rxlog	
🔁 ADD 🗸	► RUN	All_tests	•	Maintenance mode
X 🛛 🗂 🕯 9	🥲 🛅 Data source			
Item			Data bir	iding / iterations
🛛 📄 Authentica	tion - Test suite			

We can rename the Test case (i.e. from TestCase to whatever describes it). We then add a Setup and a Teardown section and add the OpenBrowser and CloseBrowser modules, to each section respectively. Opening the OpenBrowser module, by double-clicking on it, will allow us to set the URL to be used.



We can then use Ranorex Recorder to create a "module" (i.e. a set of sequential actions and/or validations on UI elements).

This will be used to implement interactions with our web site, without having to code.



We choose "Record" and then we're redirected to the browser, where we can perform actions which will be recorded.

🗘 Adı	id new action 🔹	× 0	(*)	19	@ ▲	₩ [3	• Turbo r	node 📷	Screensho	it .							
2													omment				
) Adı	d new item + 6	🔒 Track	18 0	00	≅ 9	61	R E M	tyTest1Rep	iository •	Variable	15 🎻 Cli	eanup		 Search	1 (F.	3)	
en			194 (Path					c	es 🍼 Cli	eanup		 Search	1 (F.	3)	
en 3 🕑	ApplicationUr	nderTest	18 (Path Base: /c	lon(@dd	omain='ro	botwebder	no.herokua	c app.co		eanup		 Search	1 (F.	3)	
		nderTest	% [Path Base: /c	lon(@dd	omain='ro		no.herokua	c app.co		eanup		 Search	1 (F.	3)	
	ApplicationUr	nderTest	186 (Path Base: /c	lon(@dd	omain='ro	botwebder	no.herokua	c app.co		eanup		 Search			

While recording the actions of our module, we can remove some that may be added by accident for example. We can also pause and stop recording.



In order to implement a test, we need to add at least one validation.

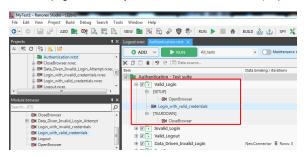
For that, we choose "Validate" and move the mouse hover the element we want to validate and click on it.

Ø H1	Fag 'Welcome Page'						
Login succeeded. Now y Relew Gape w Fleron Location Location Location Independent Indepen	nknown ekebelement, hitag chroimeweb nr. 443, 194 10, 37 False False K. Welcome Page						
			Ranor	ex Recorder (S	hift+Esc to stop)		
			13 8	🖽 Key sequen	ce {Escape down	1}	
			12 (3 Mouse	Left	Click	
			11 6	9 Mouse	Left	Click	
			10 6	Mouse	Left	Click	
				Image based re	ecording (Enable hotkeys (Ð
				VALIDATING	Validate	Continue St	op (
C 🔺 Inseguro robotivebidemo.herokuapp.o	om/welcome.html	Validation		VALIDATING	Validate	Continue	
C 🔺 Inseguro robotiveltidemo.herokuapp.c	en/wekome.html	C Refresh	element				op (
C 🔺 Inague soboreddanoleniusgo	en/wekcome 2000		element at Seksio- Den Tues Den T	Affinition of the encoders () for the second	Tennes Oudi Une Ge and Environnet Revyos cen 7 del d' neutro del puscos Revyos cen 7 del d' neutro del puscos Revyos cen 7 del d' Revyos cen 7 del	_	3@>

A validation dialog enable us to perform multiple asserts at the same time, on a given element (e.g. element exists, is visible, contains a given text). Each assert/validation will be created as a "Validate" action in our module.

Validation			۵	٠		3
Validation settings					X Studio)
Attributes Image						
Attribute Name	Attribute Value					
General						-
V Exists						
NotExists						
Enabled	True					
HasFocus	False					
C Valid	True					
Visible	True					
WebElement		 				-
Class						
ContentEditable						
ContextMenu	False					
Draggable	False					
	raise					
✓ InnerText	Welcome Page					
SafeClass	welcomerage					
TagName	h1					
TagValue						
H1Tag						
Align						
< <u>P</u> revious				<u>о</u> к	<u>C</u> ancel	

A possible "Valid Login" test case could be composed of a setup section to open the browser (i.e. using the "OpenBrowser" module), a recorded module where we enter the credentials and validate the welcome page, followed by the close browser instruction (as part of the CloseBrowser module).



The "Login_with_valid_credentials" module could be composed of these actions, depending on the actions you defined earlier.

			n.ntst Login,with,vali		UK X			
≂ K C % \$ S		RECORD	RUN RECORDING				🛰 VARIABLES 🕞 SETTINGS 🎾	sud
B-BI CloseBrowserzame		Add new action + 🔀	000000	6. V 26 Ti	arbo mode 🛛 🗖 Sch	enshot		
 BH Data_Diven_Invalid_Login_Attempt.rx B- BH Login_with_invalid_credentials.orect 	INC	# Action					Comment	
 B* Login, with, invalid, credentials orect B* Login, with, valid, credentials.come 	1.00	C Mease	Click	Left	Relative	UsernameField		
R-84 Lopput.tyrec		EI 2 Key sequence	deme			O UservaneField		
- MyTestLoting	- 1	C 3 Mease	Click	Left	Relative	Q Loginform		
B T MyTest1Repository.corep	- 14	C 4 Mouse	Click	Left	Relative	49 PasswordField		
B-SM OpenBrousec.comc		ELS Key sequence	mode			49 PasswordField		
		C Mouse	Click	Left	Relative	49 LoginButton		
		Validate	AttributeEqual	Voible	True	43 WelcomePage		
erch (F3)	Q							
NyTest1	-						-	
Groups E- Modules		*						
- BH CloseExpyroan		🔿 Add new item + 🔍 Trac	- × 0 🗖 =	9 🤋 🗖	MyTert1Reposit	ory - Variables 🎻 Cleanup	p III - Search (F3)	
8 DM Data Diver Javalid Login Attempt DM Login with invalid credentals BM Login with valid credentals		Een ApplicationUnderTen WwiczeneFage			n = Yobotwebdens.h			
EN OpenBrowser							No item selected	
		¥1						

We may decide to implement additional tests, reusing existing modules.

As an example, a test case that checks the valid logout procedure (i.e. "Valid_Logout") can use the "Login_with_valid_credentials" module as the first macro step, before executing the module and its actions that perform the logout and verify its result ((i.e. "Logout" module).

MyTest1 - Ranoves Studio - (3204) File Edit View Project Build Debug Search Teo			- d ×
		BURD 💩 🎂 SPY 💃 🏥 Default layout - 🔎	
Rojects 비 x 자 (종) 전 (원) (종, 100 승규는 Reports · · ·	resultandog Programus Authentication.ntst X. MyTestifie O ADD V > SUN All_tests	prostory.mmp Data_Driven_Invalid_Login_Attempt.comc Login_with_valid_co vMentervance mode	edentials.nee: Logout.nee: U MANAGE DATA SOURCES X Studio
Appoint A	Image: Second	faa Data binding / Resilient Devogetion	ect_ (7) P
Modula brevere (LX (carebus (C)) (C) B Tem My First (C) B M Obselvember B M Obselvember B H Login with inside properties B Login with inside properties B Login with inside properties			

Running the tests

Before running the tests, you have to build the current project which will produce a executable file. Tests are in fact performed by this executable file and not by Ranorex Studio itself.

Running tests using Ranorex Studio

Test cases can be "run" from within the Ranorex Studio UI; a build happens in the background, if needed.

NyTeitt - Ranorex Studio - (3268) File Edit View Project Build Debug Search Tr					0
or o 🔛 🖉 aco 📭 🕫 B₂ 🛱 B₂	. view 🖿 📉 🔄 🖉 🖗 🕬 🕨 🕨 🔳 🕸	6 BURLD 💩 🎂 SPY 💃 🕮	Default layout - D		
	Kathentication.ntst X MyTestSTepository.orep Data_Div				
n ∰ C % \$, ₩	O ADD V NUN All_texts	• 🔘 Maintenance mode		DATA SOURCES	Studio
8-8M Closelinowser.com	🗶 🗇 📄 🛢 🦻 🖓 🖓 🔯 Date source		Se	ach (F3)	P
BH Data Driven Invalid Login Attempt one: BH Login, with invalid credentials over BH Login with valid credentials.com	Zen Zen Authentikation - Test suite	Data binding / Relations	Description		
B-BH Logoutrinec	8 2 Valid Login 8 2 Valid Login				
- The MyTextLosting	8 2 Valid Logout				
E Mylest1Repository.corep	R R Data Driven Invalid Login	NewConnector # Rows 3			

Whenever running, Ranorex Studio uses a run configuration to know which test cases to run; in the previous screenshot it is called "All_tests" and contains all test cases, since they're selected.

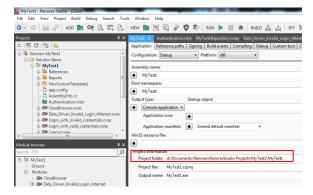
In the properties of our test suite we can customize the report file name, the reports directory, and enable the JUnit XML report which Xray can later on process.

enfication properties		- 0 -X
Authentication		
eneral Global parameters Report		
Report settings		
Report file directory: Reports		
Report file: Neutranlog		
Auto save interval: 30 Seconds (0 = CR1)		
Timustamp: Relative to test suite start time		
Counts a Junit competition copy of the import Counts angle report FOF		
Report template		Help
	Create custom template Choose custom template	
Tracing sovershots	Create custom template Choose custom template	

Running tests from the command-line

Tests can be run from the command-line, by calling the executable built by Ranorex Studio.

The execute can be found inside the project folder (obtainable by looking at the project properties), either in the bin\Debug or bin\Release folder.



To run the tests, we execute the file and pass some arguments to enable the JUnit XML report, customize the report file base file name; (use /help to find available options).

example of a shell script to run the tests MyTestl.exe /junit /reportfile:results

In this case, the report will be stored in the current directory and will be named <code>results.rxlog.junit.xml</code>.

Integrating with Xray

In order to have visibility of our test automation results in Jira, we need to generate a JUnit XML report whenever running the tests, that can then be submitted to Xray as shown in the previous section.

To submit the report to Xray, we can use our favourite CI/CD tool or a simple script for that.

Once you have the report file available you can upload it to Xray through a request to the REST API endpoint for JUnit. To do that, follow the first step in the instructions in v1 or v2 (depending on your usage) to obtain the token we will be using in the subsequent requests.

```
curl -H "Content-Type: application/json" -X POST --data '{ "client_id":
   "CLIENTID","client_secret": "CLIENTSECRET" }' https://xray.cloud.getxray.
   app/api/v2/authenticate
```

Once you have the token we will use it in the API request with the definition of some common fields on the Test Execution, such as the target project, project version, linked test plan, etc.

```
curl -H "Content-Type: text/xml" -X POST -H "Authorization: Bearer
$token" --data @"results.rxlog.junit.xml" https://xray.cloud.getxray.app
/api/v2/import/execution/junit?projectKey=XT
```

Sample batch (.bat) script to import results to Xray

```
@echo off
set client_id="DA2258616A5944198E9BE44A9A000000"
set client_secret="
5baelaa5b49e5d263781da54ba55cc7deebd7840c68fe2fdfd2a077768000000"
set project_key="XT"
set report_file="Reports\results.rxlog.junit.xml"
```

```
set jira_base_url="https://xray.cloud.getxray.app/api/v2"
```

```
for /f %%i in ('curl -H "Content-Type: application/json" -X POST --data "{
  \"client_id\": \"%client_id%\",\"client_secret\": \"%client_secret%\" }"
  https://xray.cloud.getxray.app/api/v2/authenticate') do set token=%%i
  rem echo %token%
  curl -H "Content-Type: text/xml" -X POST -H "Authorization: Bearer %
  token%" --data @"%report_file%" "%jira_base_url%/import/execution/junit?
  projectKey=%project_key%"
```

Sample PowerShell script to import results to Xray

```
try {
 $client_id = 'DA2258616A5944198E9BE44A9A000000'
 $client_secret =
'5baelaa5b49e5d263781da54ba55cc7deebd7840c68fe2fdfd2a077768000000'
 $project_key = 'XT'
 $report_file = 'results.rxlog.junit.xml'
 $jira_base_url = 'https://xray.cloud.getxray.app/api/v2'
 $json = @"
 {
  "client_id": "$($client_id)", "client_secret": "$($client_secret)"
 }
"@
 $uri = "$($jira_base_url)/authenticate"
 $res = Invoke-WebRequest -Uri $uri -Body $json -Method POST -ContentType
"application/json"
 $token = $res -replace '"',''
 #write-host $token
 $fileContent = Get-Content -Path $report_file -Raw
 $uri = "$($jira_base_url)/import/execution/junit?
projectKey=$($project_key)"
 $res = Invoke-WebRequest -ContentType "text/xml" -Uri $uri -Body
$fileContent -Method POST -Headers @{"Authorization" = "Bearer $token"}
 write-host $res
}
catch {
write-host $_.Exception.Message
}
```

After submitting the test automation report, a Test Execution will be created in Jira, containing the results for each Test case.

A Test issue will be auto-provisioned, unless it already exists, per each Test Case.

Projects	/ 関)	(ray Tu	itorials / 📘 XT-	-134							
Execu	ition	res	ults [1624	96403962	8]						
Ø Atta	ach		reate subtask	C Link issue	× [O Tests					
Descripti	on										
Add a de	scriptio	n									
Tests											
Add Te	sts 🗸										
Overa	ll Exe	cutic	on Status								
6 _{pas}	SED									TOTAL TE	STS: 6
		Filters	· •					10	~	Columns	*
	Rank *	Key	Summary ÷			Test Type≎	Dataset	Status 0			
	1	XT- 125	Valid_Login			Generic		PA	SSED		≡Ľ
	2	XT- 126	Invalid_Login			Generic		PA	SSED		∃E
	3	XT- 127	Valid_Logout			Generic		PA	SSED		∃E
	4	XT- 135	Data_Driven_Inv	valid_Login_DataI	teration_1	Generic		PA	SSED		∃E
	5	XT- 136	Data_Driven_Inv	valid_Login_DataI	teration_2	Generic		PA	SSED		∃E
	6	XT-	Data_Driven_Inv	valid_Login_DataIt	teration_3	Generic		PA	SSED		≣Ľ

The Test Suite name along with the Test Case name will be used as unique identifier for the Generic Test that will be created.

In the Test Run details of the corresponding Test in Xray, we can see this information along with the overall result. The original Test Suite name appears also as a context.

	_				
XeyTeerie / TextSection 87-134 Valid_Login	/ Test 97-125				Tex 1 of 6 O O O
Excertise States	Stantol Do 2004/2022 1254 AM Rebited On 2004/2022 1254 AM		Autypes Singlo Traine Executed By Singlo Fraine	Bardoni - Residen -	Fart Endersoneeds
+ Risdiege O					
Test details					
• Test Issue Links 😑					
Oefinition Automication Valid_Login					
- Results O					
Cardest Texture Authentication		Output -			Duration Status

When a test case fails, the corresponding Test Run in Xray will show details about the repository item that failed; its full identifier includes the name of the repository followed by the hierarchical location of the repository item.

haosan / Tentisection 37-114 / lid_Login	Text 37-125				Execute with Exploratory App.	Import Execution Result
dan latan KALO V	Stanled Dn 20(Un/2022) 00:30 PM Fielded On 20(Un/2022) 00:30 PM		Analysee Silegia Fraine Descalad By Silegia Fraine	Territori - Erithian	Test Environment	unia.
indings O						
est details errora						
Test lasse Linka \\ Definition						
Authentication Valid Login						
Recults O						
Centeut		Output			Duration	Status
TeelSuite Authoritorion		The test runs has even as "Under Articles" VM- match the specified vision (Inder the Arten Na- Yohne) (Elementari Vertretenbertenbertenbertenberten- bertenbertenbertenbertenbertenbertenbertenberten- bertenbertenbertenbertenbertenbertenbertenberten- bertenbertenbertenbertenbertenbertenbertenbertenberten- tietenbertenb	The Composition Difference Page Weiner (19) (4) (4) (4) (4) (4) (4) (4) (4) (4) (4	Page: No stemant transf to path imme Page2 is they december on NAML specification and the stemant on NAML specification and the stemant of the stemant specification and the stemant speci		MALAD

The failed element was in this case the "Welcome Page" header, in the Welcome Page, identified by "My Test1Repository.WelcomePage.WelcomePage".

jie Edit View Project Build Debug Search I	ools <u>W</u> indow <u>H</u> elp		
3 🗸 O 🔛 🖉 ADD 💽 😋 📴 🗈	væw 🖿 🗟 🛱 🔗 🦁 🎭	RUN 🕨 🔳 🎆 🛛 BUILD 📩 📩 🛛 SPY 🏌	Default la
	K MyTest1 Authentication.ottst MyTest1	Repository.nzep 🗙 Data_Driven_Invalid_Login_Attempt.c	rrec Login_with_v
= 토 C 맘 🚖 🐸			
Solution MyTest1	ADD - C INACK		
🖻 📔 Solution Items	🕽 🗶 🗇 🗂 💼 🔊 າ 🗔 🛏 Varia	bles 🎻 Cleanup	
🕁 🔤 MyTest1	Item	Path	Comment
References	ApplicationUnderTest	Base: /doml@domain='robotwebdemo.herokuapp.co	
Reports		.//div(#'container1/h1)@innertext="Error Page"	
• NewCustomTemplate1	CoginFailedInvalidUserNameAnd	OrP .//div(#'container')/p(@innertext>'Login failed. Invalid .	
- C app.config	Q Logout	.//div[#'container]/?/?/a[@innertext='logout']	
 AssemblyInfo.cs 	- O LoginPage	.//div[#'container]/h1]@innertext='Login Page]	
- Authentication.rxtst I	Container	.//div[#'container']	
Oliveration CloseBrowser.nxrec	⇔ SomeBodyTag	body	
	WekomePage	Base: /dom@domain='robotwebdemo.herokuapp.co	
B-OM Login_with_invalid_credentials.nxrec	↔ UsernameField	.//input]#'username_field]	
	Control C	.//div[#'container']/form[@name='login_form']	
Of Logout.rxrec	Passwordnield O LoginButton	.//input]#'password_field] .//input!#'login_button1	
🖧 MyTest1.cd	O WelcomePage X EDIT IN S		
- MyTest1.rxtmg	H E Explorer	Base: /menubar[@processname='explorer']	

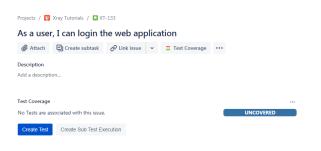
Tips

Seeing the impacts of test automation results on user stories or requirements

After uploading the test automation results, users can link the Test issues to existing user stories or requirements. That will enable users to track coverage and thus assess if user stories are covered by automated test scripts and if based on that, the corresponding user story can be considered to OK or NOK.

Please remember that coverage is an heuristic but that can still be quite helpful to assess the readiness of user stories, individually or at the release level.

Assuming we have a user story (new or existing), we can then link it to the Tests that correspond to the Test Cases implemented using Ranorex.



We can do that right from the user story issue screen, using the "Link issue" action and the "is tested by" issue link . Then, we select the Tests that were auto-provisioned earlier on upon the first import of the test report.

Projects / የ Xray	Tutorials / 🖪 X1	-133				
As a user, I c	an login tl	he web applic	ation			
🖉 Attach 🛛	Create subtask	🔗 Link issue 👻	Test Coverage	•••		
Description Add a description						
Linked issues						
is tested by	~ O	XT-125 × XT-126				0
		N ISSUES XT-126 Invalid_Login				
Test Coverage	XT	-126 (Exact key)				
No Tests are associat	ed with this issue				UNCOVERED	
Create Test C	reate Sub Test Ex	ecution				

Finally, on the Test Coverage panel we can see the latest test results right from the user story issue screen along with the calculated coverage information for the user story.

As a user, I can login the web application	
Description Add a description Linked issues is tested by XT-126 Invalid_Login	
Add a description Linked issues is tested by XT-126 Invalid_Login = TO Dow	
is tested by Image: XT-126 Invalid_Login TO Down	
	+
O XT-125 Valid_Login	
Test Coverage Create Test Create Sub Test Execution	
Calculate the Test Coverage for the following scopes.	
Latest Version Test Plan	
Test Environment All Environment OK	
C Final statuses have precedence over non-final.	_
Status ⇒ Key ⇒ Summary Test Status ⇒	
TO DO XT-125 Valid_Login PASSED TO DO XT-126 Invalid Login PASSED	

Any additional imports of results, will appear automatically reflected on the user story issue screen as the Tests are already linked to the user story.

Run iterations and data-driven tests

Ranorex Studio has support for run iterations and data-driven tests.

These are two different concepts; while run iterations are just a way to run the same test case multiple times by executing the exact same modules and actions, data-driven tests will impact on the action being performed (e.g. for exercising the same test case but with different inputs).

It's possible to have visibility of the corresponding test results in Jira using Xray but some care should be taken.



Due to the way Ranorex Studio reports these results on the JUnit XML report, different Test issues will be created for each run iteration or data row.

We should have this in mind as it hardens management of these Test issues (e.g. number of unrelated Tests, linkage to user stories).

Run iterations

On the properties of a test case, we can configure the number of run iterations (i.e. iteration count).

Logout.nrec Authentication.ntst × Login_with_valid_		Properties	#X
O ADD V NUN All_tests	Maintenance mode	12 111 Misc	
🛠 🗊 📋 🛢 😕 🤉 🖓 🕅 Data source		Checked	True
Iten	Data binding / iterations	Description	
Authentication - Test suite		Error behavior	Continue with sibling
B V Valid Login	2 Iterations: 3	Guid	e09e310c-bd52-4030-99b2-bbabcd7e
	Prevations: 3	Iteration count	3
B (SETUP)		Name	Valid_Login
OpenBrowser		Report level	Inherit
 D4 Login_with_valid_credentials 		Retry count	0
E- [TEARDOWN]		Warn for unbound variable	
■ CloseBrowser		1	

After running the test case and importing the results to Xray using the JUnit XML report, a Test Execution with 3 Tests is created in Xray.

Projects / 💱 Xray Tutorials / 🖪 XT-150

Execu	tior	n result	s [1624	97465863	3]						
Ø Atta	ich	Create	subtask	🔗 Link issue	×	O Tests					
Description Add a des		on									
Tests											
Add Te:	sts 🗸										
Overal	l Exe	cution S	Status								
2											
3 PASS	ED									TOTAL TE	STS: 3
	,	Filters v						10	~	Columns	
	,		Summary 0		Test Type≎	Dataset	Status =	10	*	Columns	
	,	Key		RunIteration_1		Dataset	Status :		~	Columns	*
	Rank	Key: XT- 151	/alid_Login_F	RunIteration_1 RunIteration_2	Type : Generic	Dataset		D	~	Columns	✓ Actions
J pass	Rank 1	XT- 151 XT- 152	/alid_Login_F /alid_Login_F		Type : Generic Generic	Dataset	PASSE	D	~	Columns	✓ Actions

Due to the way run iterations are reported in the JUnit XML report, each run iteration for our test case is abstracted as a different Test issue, with the run iteration being part of its definition.

In other words, we'll have as many Test issues as the iteration count configured for the test case.

Xoy Tutoria: 7 Ten Decutor XT-15 Valid_Login_Runiteratio					Text Lot 2 O O O Decode with Depictetory App
Constant Salata	Sarrol (In 28/sar/2005 00:53 PM Histohed Ox 28/sar/2003 00:55 PM		Ausignee Salagie Protee Konstand Ny Salagio Franke	Venium - Favition -	Text Environments
 Findings O Test details manual 					
 Definition Authentication Valid_Login_Date Results 0 	ribeston.)				
Context TextSalts Authentication		Output			Dervice Status

Data-driven tests

To make a data-driven test we need to use some variables in our actions, instead of using hardcoded values/strings.



Then we need to configure a data-source for the selected test case.

Logout.nrec Authentication.ntst* X Login_with_valid_cred	entials.nmec results.ndog Data_Driven_Inva	id_Login_Attempt.nzec		
ADD V NUN All_tests	Maintenance mode		MANAGE DATA SOURCES	🎾 Studio
🛠 🗇 🖺 🛢 🦻 🤗 🥙 📴 Data source		5	Search (F3)	P
Item	Data binding / iterations	Description		
🗆 📴 Authentication - Test suite				
🗷 📰 💽 Valid_Login				
🗷 🛅 💽 Invalid_Login				
B 🔲 💽 Valid_Logout				
🖲 📄 Data_Driven_Invalid_Login	NewConnector			

There are different types of data-sources; a simple built-in data table can be used to specify several named columns and some rows of data for them.

lanage data s	ources				×	
New	- @ <u>0</u>	lone 📋	<u>D</u> elete			
	e data table		 Connector type 		Use count	1
	onnector		SimpleDataConr	ector	1	
	connector					
SQL c	onnector					
Data sourc	e config					
Name:	NewCor	nnector				
	Auto	-load data when	test suite opens			
Mask:	(No col	umn selected)			•	
Label:			eration count will be us		•	
	Select a	column to use as	s a label for data iterati	ons in the report		
				<u>O</u> K	Cancel Apply	
				<u>O</u> K	<u>Cancel</u> <u>Apply</u>	
ata_Driver	_Invalid_l	Login prope	rties	<u>O</u> K	<u>Cancel</u> <u>Apply</u>	
_			rties lid_Login	QK	<u>Cancel</u> <u>Apply</u>	
_					<u>Cancel</u> <u>Apply</u>	
_		ven_Inval			<u>Cancel</u> <u>Apply</u>	
Da	ata_Driv Data sou	ven_Inval	lid_Login	on	<u>Cancel</u> <u>Apply</u>	
Da General New	ata_Driv Data sou	ven_Inval	lid_Login inding Conditi or - SimpleData	on	<u>Cancel</u> <u>Apply</u>	
General New	Data sou	ven_Inval rce Data b lewConnect password	lid_Login	on	<u>Cancel</u> <u>Apply</u>	
▶ Da General New u ▶ 1 dr	Data sou	ven_Inval rce Data b lewConnect password random	lid_Login inding Conditi or - SimpleData	on	<u>Cancel</u> <u>Apply</u>	
▶ Da General New 1 2	Data sou Data sou I M sername ummy1 ummy2	ven_Inval rce Data b lewConnect password random whatever	lid_Login inding Conditi or - SimpleData	on	<u>Cancel</u> <u>Apply</u>	
General New 1 dr 2 dr 3 dr	Data sou Data sou T N sername ummy1 ummy2 ummy3	ven_Inval rce Data b lewConnect password random	lid_Login inding Conditi or - SimpleData	on	<u>Cancel</u> <u>Apply</u>	
General New 1 dr 2 dr 3 dr	Data sou Data sou I M sername ummy1 ummy2	ven_Inval rce Data b lewConnect password random whatever	lid_Login inding Conditi or - SimpleData	on	<u>Cancel</u> <u>Apply</u>	
General New 1 dr 2 dr 3 dr	Data sou Data sou T N sername ummy1 ummy2 ummy3	ven_Inval rce Data b lewConnect password random whatever	lid_Login inding Conditi or - SimpleData	on	<u>Cancel</u> <u>Apply</u>	
General New 1 dr 2 dr 3 dr	Data sou Data sou T N sername ummy1 ummy2 ummy3	ven_Inval rce Data b lewConnect password random whatever	lid_Login inding Conditi or - SimpleData	on	<u>Cancel</u> <u>Apply</u>	
General New 1 dr 2 dr 3 dr	Data sou Data sou T N sername ummy1 ummy2 ummy3	ven_Inval rce Data b lewConnect password random whatever	lid_Login inding Conditi or - SimpleData	on	<u>Cancel</u> <u>Apply</u>	
General New 1 dr 2 dr 3 dr	Data sou Data sou T N sername ummy1 ummy2 ummy3	ven_Inval rce Data b lewConnect password random whatever	lid_Login inding Conditi or - SimpleData	on	<u>Cancel</u> <u>Apply</u>	
General New 1 dr 2 dr 3 dr	Data sou Data sou T N sername ummy1 ummy2 ummy3	ven_Inval rce Data b lewConnect password random whatever	lid_Login inding Conditi or - SimpleData	on	<u>Cancel</u> <u>Apply</u>	
General New 1 dr 2 dr 3 dr	Data sou Data sou T N sername ummy1 ummy2 ummy3	ven_Inval rce Data b lewConnect password random whatever	lid_Login inding Conditi or - SimpleData	on	<u>Cancel</u> <u>Apply</u>	
General New 1 dr 2 dr 3 dr	Data sou Data sou T N sername ummy1 ummy2 ummy3	ven_Inval rce Data b lewConnect password random whatever	lid_Login inding Conditi or - SimpleData	on	<u>Cancel</u> <u>Apply</u>	
General New 1 dr 2 dr 3 dr	Data sou Data sou T N sername ummy1 ummy2 ummy3	ven_Inval rce Data b lewConnect password random whatever	lid_Login inding Conditi or - SimpleData	on	<u>Cancel</u> <u>Apply</u>	
General New 1 dr 2 dr 3 dr	Data sou Data sou T N sername ummy1 ummy2 ummy3	ven_Inval rce Data b lewConnect password random whatever	lid_Login inding Conditi or - SimpleData	on	<u>Cancel</u> <u>Apply</u>	

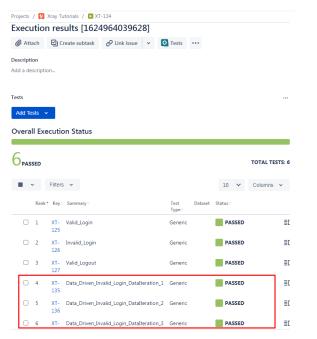
We then need to bind the columns of our data-source to the variables used in the previous module.

Data_Driven_Invalid_Login	
rel Date source Data binding Condition	
ud variabiles to data sources	
ata-source column	Variable
	Data, Diven Jinvalid, Login, Attempt.usemame
	Date, Diven Javalid, Login, Attempt personnd

We can run the test case in Ranorex Studio and see the results for each data row.

Computer/Endpoint SERGIO-PC		Execution time 25-06-2021 11:19:02	Test case result summary
Operating system Windows 7 Service Pack 1 64bit		Screen dimensions 1366x768	
OS Language en-US		Duration 11.97s	3x Success
Total errors 0		Total warnings 1	
Expand test containers	Expand details	Collapse all	
Expand test containers Test container filter:			
Test container filter:	Success 🗹 Faile		nodules, please.
Test container filter:	Success 🗹 Faile	ed Blocked	
Test container filter: O Warnings occurr • Data_Driver	Success I Faile	ed Blocked formation see the report of the individual m Rows: 3	nodules, please. 11.5% 5.7%
Test container filter: O Warnings occurr C Data_Driver	Success Failed. For additional in Invalid_Login	ed El Blocked formation see the report of the individual m O Rows: 3 Data Rows: 1	11.95

After running the test case and importing the results to Xray using the JUnit XML report, a Test Execution having 3 Tests related to our data-source is created in Xray.



Due to the way run iterations are reported in the JUnit XML report, each run iteration for our test case is abstracted as a different Test issue, with the run iteration being part of its definition.

In other words, we'll have as many Test issues as the iteration count configured for the test case.

Thereof De 20(ton/2021 12:54 AM Robust De 20(ton/2021 12:54 AM		Analysee Silegia Fraire Executed by	Venium - Breision	Test Trainsvents	
Enlabed Dr.		Constant By			
			Envision		
29(504)2021-11.64 AM					
		Sangia Reize			
ogie_Databeation_1					
	Oveput			Duration	Status
	oge_Deabhrace_1		,		

Ranorex's built-in integration with Jira

Ranorex Studio has a built-in integration with Jira. It can be used, for example, to open bugs, resolve, and reopen them depending on testing results.

The built-in Jira integration is totally independent from the Xray integration described in the current article and may be eventually complementarily.

References

- Ranorex web site
- Ranorex User Guide
- Ranorex vs Selenium WebDriver .
 - Integrating Ranorex with Jenkins
 - blog post
 - documentation
- Overview
- Ranorex concepts and mapping to Xray
- Prerequisites
- Implementing automated tests
- Running the tests
 - Running tests using Ranorex Studio