RANOREX

WEB AND MOBILE TESTING USERGUIDE

TABLE OF CONTENTS

WEB AND MOBILE TESTING	3
WEB TESTING	4
Build a web test	5
Website structure in Ranorex Studio	
Advanced web testing	20
Cross-browser testing	27
Endpoints	44
Endpoint settings	
Add an Android/iOS endpoint	
Add a WebDriver endpoint	51
Capabilities configurator	58
Ranorex Parallel Runner	59
Mobile Testing	61
PREPARE YOUR DEVICE	62
Prepare your device	63
Update Service Instrumentation Wizard	66
Connect mobile devices	75
Instrument apps	78
Create a mobile test	85
Run a mobile test	96
Android app testing example	98
Android mobile web test example	
iOS app testing example	
iOS mobile web test example	
Advanced mobile testing	130
iOS Service App	
Non-UI Testing	
Cross-device mobile tests	
Automate Android system apps	145
Test Android Wear apps	
Instrumentation with source code on iOS	

Web and mobile testing

Web testing and all different types of mobile testing represent an emerging field concerning automated software testing. Ranorex offers a wide range of tools and methods to cover as many test challenges as possible. They are all described herein.



Web testing

This chapter describes how to test web applications and web sites using Ranorex Studio. Thanks to the Ranorex plug-ins for Microsoft Internet Explorer, Mozilla Firefox, Google Chrome, and Chromium, building and running web tests is essentially the same as doing so for desktop tests. However, there are some differences and important considerations. We'll explain these in the various subchapters.

Supported browsers

Ranorex Studio supports the following browsers for test automation:

- Microsoft Internet Explorer
- Microsoft Edge (Chromium-based)
- Microsoft Edge (Legacy)
- Mozilla Firefox
- Google Chrome
- Chromium

Ranorex Studio uses special browser plug-ins to enable test automation with these browsers.

🚹 Note

You can test even more platform and browser combinations with our → Selenium WebDriver integration.

Web testing overview

Web testing and desktop testing in Ranorex Studio are very similar. The basic web testing process in Ranorex Studio is as follows:



Note

a

Web sites can have highly complex structures. This is why it may sometimes be a good idea or even necessary to extend the above process by manually tweaking the recorded test with additional code. We'll cover this in the later subchapters.

Recommended knowledge

Before you start web testing, you should be familiar with the basics of how web sites work. You should also understand the concepts covered in the following Ranorex-Studio chapters:

Ranorex Studio fundamentals

The ---> Ranorex Studio Fundamentals contain the chapters dealing with all the basic elements of working with Ranorex Studio, like actions, repositories, and reporting.

Ranorex Studio advanced and expert

For web testing, we recommend reading through the advanced chapters → RanoreXPath and → Ranorex Spy. Also, knowing your way around → code modules and the → user code library will be helpful when tweaking web tests with additional code.

Build a web test

In this chapter, you'll learn how to build a simple web test from start to finish, including recording the test, running it, and viewing the test run report.

Test definition

Our test will contain the following steps:



Start the browser and open the URL www.ranorex.com/web-testing-examples/

Check if the Ranorex Studio logo is displayed at the top.

	22 R.	anore	x Studio	- Web test page 🤇	+	-			×
€	\rightarrow (C	ŵ	0 🖲 🦀	https://www.ranorex.com/web-testing-examples/#	111		к	≡
			Rx	Ranore	Studio	u 4	Mens	15	~
	ſ			J	Ranorex Studio - Web example page				
	Ľ			10	This website is used to test the Ranorex Studio features.				
	Ľ			1 .	Simple link (new window)				
	-		L		Link with title Open dialog				

Click Open dialog.

3

4

Click OK to confirm and close the dialog window.

J	Ranorex Stud	io - Web exa	mple page	
(* (* - (* - -• (* :	This website is used to test Simple link Simple link (new window) Link with title Open dialog	st the Ranorex Studio fe	atures.	
<pre>cont < 2) { cont < 2) { deta.\$inage.outer deta.\$inage.outer </pre>	ROWS 1 1 2	NA Fin €e	ink clicked!	JE
and impriliethy	3	ты	4 ок	



🚺 Note

In automated tests, using the action **Close application** is usually more robust than closing an application "manually" by clicking the X in the application window. This is why the default web test project uses this action.

Create a new web test

Click Web.

We'll create a new web test using the ---> RocketStart wizard.



3

5

On the Ranorex Studio Start Page, click New test solution using wizard... or go to File > New > Solution using wizard...

Follow the instructions of the wizard. Name your solution **WebTest**. When prompted, select Mozilla Firefox as your browser (make sure it's installed on your computer) and specify www.ranorex.com/web-testing-examples/ as the URL.



4 When asked to select the recording behavior, **select Add browsers to whitelist**, as our test will only cover interactions with the previously selected browser. If we were going to interact with desktop applications and wanted this in our test, we'd select **Do not use whitelisting**.

On the final screen, **click Finish**. Ranorex Studio will then open the prepared solution.

Create your new solutio	an .	ล	×		
Basic data	Web application	Configure behavior Fins			
	Select recordi	ng behavior		E	
	How do you want Ranorex Stud	io to behave while recording?			· · · · ·
Add br	owsers to whitelist	Do not use whitelisting		b application Configure Almost there	behavior Finish
	0 4	Ē		lick Create to create your solution a	and get started in Ranorex Studio.
				Ranorex Studio.	
		Show all process including the supported browsers		he tutorial panel. It will guide h becoming familiar with the	
Hint: Use the whitelis	iting feature of Ranorex Studio t	o add or modify entries later.			The tutorial panel on the right wi guide you through the next step
		+ Back	Continue +		
		10 CAS			Glad Fini

The prebuilt web test will look like this:

💃 WebFeit - Ransee Studie		- O X	
● ● ■ ■ ■ = = = = = = = = = = = = = = =		tiquet - D	
nijen AX	WebTest.nxtst X	z Pototęcin Fanora Studio 🛛 🕹 🗙	
Studier WebTert Solution berry	X C C B 8 9 9 10 00 00000. Devision 0.000 000 000 000 000 000 000 000 000	💃 Webfet-Fanses Stots För Fåk Kox Pojot Rald Désig Sarch Taol	ing Mindow Male
By WebFert B - MebFert B - MebFert C, any config C, Any config C, Any config	☑ 📄 WebTest - Test suite	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	Recording1.rxrec X 3 H = 10 K = Manature P
BP Classification on m BP Classification and BP Classification and C. Program of BP Classification on conc	🖃 🗹 💽 TestCase	I Saludian WebTerr Solution term Mit WebTerr Bit WebTerr Bit References	RECORD RUN
 Web Test symp Web Test symp 	Image: Note the N		
 #C. Metrocomplementations 	2 OpenBrowser	BP Charlington rem BP Charlington rem Def Charlington rem Control to same rem Control on the Second system Second system	O Add new action - ※ □ □ = 9 ♥ ▲
Ramon madult langua X	Recording1	WebTestoring WebTestoring WebTestoring WebTestRepeakeryonop	# Action
G By NecTez	🖮 [TEARDOWN]		
Modules MR Coolinescer MR Coolinescer MR Resenters MR Resenters	CloseBrowser	Emante module Secondar B 20 Secondar (S.S.) (2)	<u>م</u>
		C - No AvecTex - Groups D- Modules - BM Occelfrencer	
		Bi Open Secure Bi Reconting1	- In Trady
			ko item selected
	4		
Ready			
		Faady	a a German Stop T

Test suite view

1

This is where you build and control your tests.

2 The solution comes with a test suite project that contains a simple prebuilt test suite structure. It contains a test case with three recording modules: **OpenBrowser**, which starts the browser and navigates to the URL; **Recording1**, which is empty and ready for you to record test actions; and **CloseBrowser**, which closes the browser.

3 Recording module view

In the recording module view of **Recording1**, you can record and manage test actions.



Empty actions table

This is where your recorded actions appear.

Record the test

Before you start recording, make sure Firefox is running and the URL www.ranorex.com/webtesting-examples/ is opened. Don't record these steps. They are already included in the prebuilt project.

Once that's done, you can start recording.

In the recording module view, **Click RECORD**. Ranorex Studio is minimized.

The Recorder control panel appears in the bottom right.



Validate the logo



In the control panel, **click Validate**.





Mouse over the Ranorex Studio logo and **click** it when the purple frame surrounds it as in the screenshot.

Ranorex Recorder (Shift+Esc to stop)	
	Ranorex Recorder (Shift+Esc to stop)
Click in your application to start recording	Click in your application to start recording
Image based recording Enable hotkeys II	
RECORDING Validate Pause Stop Add	Image based recording Enable hotkeys ①
	→ 🔄 VALIDATING 🛛 Validate 🔴 Continue 📄 Stop 😔 Add

3 The **Select element** window opens. Check if the correct image has been selected and **click Next** to confirm.

The Validation settings open. We're checking that the logo exists and is visible. Both of these attributes should already be selected. **Click OK** to confirm.

		Validation	×
		Select element	½ Studio
Validation			_
Validation settings Attributes Image Attribute Name General Exists Visible	Attribute Value	C Refresh Overview Advanced Image: Host in the set page' General Enabled Tr Image: Imag	ue ue lie he EIR auth wrights.
Veible Veible Veible ContentEdtable ContentEdtable ContentEdtable Interiet Itidden Id Interiet TagName TagName TagValue DV:Tag Dame	True Folse Folse header-top div	Com Office Preised Com Rancer Studio - Web test page - Mozilia Firefox Com Rancer Studio - Web test page - Mozilia Firefox Com Mage MicrosoftEdge Com Mage MicrosoftEdge Com Mage MicrosoftEdge Com Neglorat Com Neglorat Com Neglorat	r-top'
< Previous			Nex - Cance

Open and confirm the dialog window

4

2

In the browser, **click Open dialog**. A simple dialog window opens.

Click OK to confirm and close the dialog.

L	Ranorex S	Studio - We	eb example page	
<u> </u>	This website is use	ed to test the Ranor	ex Studio features.	
<u> </u>	Simple link			
<u> </u>	Simple link (new v	window)		
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	Link with title			
0	<u>Open dialog</u>]		
	ROWS	NA	Link elieked!	U E
<pre>deta.\$inage.outer</pre>	1	Fig	LINK CICKED!	
acta.\$inage.outer	2	Se		n
The second s	3	Ты		



Finally, in the Recorder control panel, **click Stop** to finish the recording.



Results

The control panel disappears and Ranorex Studio comes back into focus, showing the following:

X WebTest - Ranorex Studio												-	o ×
File Edit View Project Bu	ild Debug S	Search Tools	Window Help										
O - O 🖹 📝 💾	🚰 🕴 ADD	🛤 🗣 🕒 (🖬 🚯 🛛 VIEW	D 🖭 🗄	ì 🔊 🏮		•	*		SPV 🗶 :	Default layo	ut v ,O	-
Projects		• × WebTe	st.ntst Recording	l.nrec* 🗙									÷
# 🖻 이 탭 🍰 🎽			RECORD	NUN 🚬							- VARIABLES.	_ 🖹 SETTINGS	🌠 Studio
- R Solution WebTest	🛟 Add n	ew action	• % 🗇	Č 🗎	9 Q		▼	≣le Tu	urbo mode	Screen:	hot		n
References B- Reports	#	Action											-0
	🗹 1	Validate		Attribute	Equal			Visible	2	True		<놈 Hea	derTop
E-ON CloseBrowser.rxrec	🕀 2	Mouse		Click			1	Left		Relative		😪 Ope	enDialog
- C Program.cs	🕀 3	Mouse		Click				Left		Relative		< > Son	nePTag
Rangres module browser	Add	< new item	- 🧟 Track		- 	÷	19	Q		ebTestRepo	> ository + +	Variables	2
Search (F3)	Item								Path				
Groups		Applicatio	nUnderTest						Base: /domi/	@domain=		av com'l	
Modules		<2. Header	Тор						//divi#'head	er-top'l	www.runor	calcolling	
Di OpenBrowser		Scr	eenshot1					- 1	// and - meas	ci cop j			
- M Recording 1		< <u>∠</u> OpenD	ialog						//div[#'Cont	ent']//a[@i	nnertext='0	pen dialog']	
		< SomeP	Tag						//div[#'Cont	ent']/div[1]	/p[3]		
		¢		_							3		

Action table with three recorded actions.

Repository with repository items referencing the three UI elements interacted with during the recording.

Run the test and review the report

Let's run the test and review the test run report.



1

2

Close all instances of Firefox.

Click the tab **WebTest.rxtst** to switch to the test suite view and **click RUN**.

X WebTett - Rances: Stado File Lott How Project Built Debug Search O O O R M B W W Web Projects	100k Windon Hep Test.rxtst	WebTet - Rances Studio (J.dministrato)	- E X Extent input - D	- a ×
In IN<		The Date Your Project bail date such the Company of the Date of the Company of the Project Project P	WebTest - Test suite	Erech-Fit
El Withfeetheekorg.comp	Add new teen * € Tradue K □ II I	BO Cynolinear orac Brywynt S BO Ferselwyl rwe St Welfactwa Welfactwa B 20 Welfactwar B 20 Welfactw	[SETUP] [SETUP] [Recording1 [TEARDOWN] [TEARDOWN] [CloseBrowser	

Report

Once the test run has finished, the report will appear. In our case, it should look like this:

💥 WebTest - Ranorex Studio (Administrator)			- 🗆 X
File Edit View Project Build Debug Search	Tools Window Help		
💁 o 🖹 🖬 🔐 🖓 🗛 💼 G	🖬 🖪 🖬 🔚 🗤 🖿 🖬 🛱 🔗 🤅	nun 🕨 📰 🏦 🛛 Buito 🏭 🛔	SPV 🔏 🏛 Default layout 👻 🔎
Projects	* X WebTest.ntst Recording1.nrec WebTest_W	181003_092050_nlog ×	
キ 🖻 C 🔮 🏡 🌌			
Solution WebTest Solution Items Solution Items Solution References References	Rx Ranorex Studio		
Capp.config Casemblyinfo.cs MCCoseBrowser.nurec	WebTest		
OH OpenBrowser.rxrec D Program.cs OH Recording 1 proc	Computer/Endpoint	Execution time 10/3/2018 9:20:58 AM	Test case result summary
- WebTest rxtmg - MebTest rxtst	Operating system Windows 10 64bit	Screen dimensions 5760x1200	
🛓 📰 WebTestRepository.rxrep	OS Language en-US	Duration 12.35s	Tr Success
	Total errors 0	Totel warnings 0	
	Expand test containers Expand detail	Is Collapse all	
Ranorex module browser	■ × Test container filten Success	Failed Rocked	
Search (F3)	P TestCase		
C. No Weblex			

A detailed view of the test case shows that all actions were executed successfully.

est containe	er filter:	Success	✓ Failed	Blocked	
🕶 🗈 Te	stCase				7.6s
SETU	P				1.18s
- 💽 Rec	ording1				6.295
Filter:	Info	Success			
Time	Level	Category	Message		
00:02.339	Info	Validation	Validating Attr	ibuteEqual (Visible='True') on item 'ApplicationUnderTest.HeaderTop'.	
00:06.250	Success	Validation	Attribute 'Visib	ole' of element for item 'WebTestRepository.ApplicationUnderTest.HeaderTop' does match the specified va	lue.
00:06.409	Info	Mouse	Mouse Left Cli	ck item 'ApplicationUnderTest.OpenDialog' at 42;10.	
00:07.605	Info	Mouse	Mouse Left Cli	ck item 'ApplicationUnderTest.SomePTag' at 249;13.	
▶ TEAR	DOWN				125m:



Website structure in Ranorex Studio

Ranorex Studio can recognize the entire HTML architecture of a website and make it available for automated tests. Use Ranorex Spy to browse through a website's structure and see how it's represented within Ranorex Studio. Websites are organized into three basic parts in Spy. These are explained below.

Domain object model (DOM)

This is the top-level node. Each website opened in a browser has its own DOM node in Spy.

🛠 Ranorex Spy - (64bit) - Live	-	
BROWSER REPOSITORY	🖹 SETTINGS	🌾 Studio
Q TRACK X Click X to leave edit mode		
BROWSER & RESULTS		
Eù C C Edit Mode Host Host 1200 🕑 1134:75		
Dom 'User Guide Ranorex' Dom 'User Guide Ranorex' Dom 'User Guide Ranorex' Dom 'Dropbox'		^
Dom 'Ranorex Studio - Web test page'		

Browser in Spy showing five DOM nodes, each representing an individual website.

Browser-specific elements

Each browser also has browser-specific elements that aren't contained in a DOM node. These relate mostly to control elements and application windows, e.g.:

- Window controls (minimize, maximize, close)
- Pop-ups

1

• Dialog windows

These elements are organized in browser-specific FORM nodes.





Tab control elements in a browser (note the website in the tab is still a DOM node).

Window controls in a browser.

Dialog window in a browser.

These elements are organized in browser-specific FORM nodes.

🔏 Ranonex Spy - (646it) - Live	- D X
-> BLEMENT BERCHENT	🗖 Aways on top 👔 SETTINGS 🌾 Study
TRACK X /form(@title="AHome(-\ Dropbox(-\ Google(C//container/@accessiblename="AHome(-\ Drop Cobit 1 to leave edit made Cobit 1 to leave edit made	box\ -\ Google\ C]Weutten[@accessiblename='Maximize]
BROWSER & RESULTS PATH EDITOR	
ED C E III III UIT - Edit Mode Button 'Maximize'	
Host ' Form 'Home - Dropbox - Google Chrome' Container Home - Dropbox - Google Chrome' Container 'Google Chrome' Container 'Pane' Maximize'	amic (dyn) is ten [batten] en [batten] e 8. 1, 46, 29
The index of this element in the parent's child list. Is '-T if the index is	unknown.

በ Note

Browser-specific elements can cause issues in cross-browser tests because naturally, different browsers have different elements. They should either be avoided or individualized in these cases.

You can handle browser-specific elements in the RanoreXPath or in code.

A RanoreXPath handling a button for two different browsers by way of their process names would look as follows:

```
/form[@processname='firefox' or
[@processname='chrome']//button[@text='OK']
```

If you want to include more process names, the following is a more economical way of doing so:

```
/form[@processname~'[iexplore|IEXPLORE|chrome|firefox]'//button[@text=
'OK']
```

The Ranorex Automation Library included in Ranorex Studio also makes it possible to differentiate between browser-specific elements by way of various variables and methods in

code. For example, the variable **BrowserName** allows distinguishing between different browsers. The following code example demonstrates this:

```
C#
// Click the OK button in popping up dialog of one of the supported browser
// If the current browser is Internet Explorer
if(webDocument.BrowserName == "IE")
Button okIE = "/form[@processname~'(iexplore|IEXPLORE)']//button[@text='OK']";
okIE.Click();
// If the current browser is Mozilla Firefox
else if(webDocument.BrowserName == "Mozilla")
Button okFF = "/form[@processname='firefox']//button[@text='OK']";
okFF.Click();
// If the current browser is Google Chrome
else if(webDocument.BrowserName == "Chrome")
{
Button okChrome = "/form[@processname='chrome']//button[@text='OK']";
okChrome.Click();
// If the current browser is Apple Safari
else if(webDocument.BrowserName == "Safari")
Button okSafari = "/form[@processname='Safari']//button[@text='OK']";
okSafari.Click();
```

VB.NET
C#
VB.NET
' Click the OK button in popping up dialog of one of the supported browser
' If the current browser is Internet Explorer
If webDocument.BrowserName = "IE" Then
Dim okIE As Button = "/form[@processname~'(iexplore IEXPLORE)']//button[@text='OK ']"
okIE.Click()
' If the current browser is Mozilla Firefox
ElseIf webDocument.BrowserName = "Mozilla" Then
<pre>Dim okFF As Button = "/form[@processname='firefox']//button[@text='OK']"</pre>
okFF.Click()
End If
' If the current browser is Google Chrome
ElseIf webDocument.BrowserName = "Chrome" Then
Dim okChrome As Button = "/form[@processname='chrome']//button[@text='OK']"
okChrome.Click()
End If
' If the current browser is Apple Safari
ElseIf webDocument.BrowserName = "Safari" Then
Dim okSafari As Button = "/form[@processname='Safari']//button[@text='OK']"
okSafari.Click()
End If

The WebDocument class and its variables and methods are described in the Ranorex API documentation under Ranorex > WebDocument.

Website elements

Website elements are targeted using ---> RanoreXPath, just like elements in a desktop application. The element browser in Spy displays this as follows:



In the repository, identified website elements are referenced as repository items and displayed as follows:

😻 Dropbox 🛛 🗙	+		-		×
← → C ²	pbox, Inc (US) http:	s://www.dropbox.com/?landing=dbv2 •••• Q Search	lii\ (I X	≡
Get Dropbox Basic		Compare plans · S	ign ir	ı	^
🕒 Add new item 👻 🔍 Track 🛛	< 🗇 🗋 🗎 🥬	🧟 🔄 🔄 CrossBrowserTestRepo 🔹 🛏 Variables 🝼 Cleanup	1 • §		
Item		Path			5
ApplicationUnderTest	+	Base: /dom[@domain='www.dropbox.com']	- 1		1
<>> SignIn		.//div[#'component8131148436911192318']//section/?/?/ul/?/?/a[@innertext='Sign in'	1		100
LoginEmail771864090005	00000	.//input(#'login_email7718640908696506']			
 LoginPassword037840912 	2448768416	.//input[#'login_password037840912448768416']	1		100
RememberMe925463212	0187671	.//input[#'remember_me9254632120187671']			-
LoginButtonSigninButton	ButtonPrimar	.//div[#'component6240627703226992946']//form[@action='https://www.dropbox.co	om/lo	1.00	
😑 📄 PageletTopMenuAccour	ntButtonContain	Base: .//div[#'maestro-header']/div/div[2]			
<a>SignOut		.//nav/?/?/a[@innertext='Sign out']			
<>> JohnPublic		.//nav/?/?/h2/div/div[1]/div[@innertext='John Public']			
McPopoverTrigger		?/?/button			

Advanced web testing

In this chapter, we'll cover some advanced topics regarding web testing.

Use the Wait for action to wait until an element has loaded

Loading times that take longer than usual are a common issue in web tests. They can cause tests to fail and can get quite annoying, as they are often outside of the tester's control. One way to work around these loading times is by instructing the test to wait until a certain element has loaded. In Ranorex Studio, you can easily do so with the **Wait for** action.



In the action table, **add** a Wait for action after an action that loads a web page and before an action that manipulates an element on the loaded page.

💥 WebTest - R	anorex Studio					-	0 X
Eile Edit Vie	w <u>P</u> roject	Build Debug Search Io	sols Window Help				
	8 R	a 🖉 ADD 📷 📬	🔓 🎫 🕼 VIEW 🖿 🖄 🛱 🔗	🤨 RUN 🕨 📰 🎊 BURD 🏭 🚣	SPY 🄏 📰 Default layout	م .	=
Projects			× WebTest.ntst Recording1.nrec × Web				÷
⇒ Kh C 9	1 📩 🖉		RECORD L RUN			👞 VARIABLES 🚺 SETTINGS	1/2 Studio
B-R Solution	WebTest			2. Cl. (
e R Wel	#	Action					
0 2	() 1	Wait for	Exists	10s 🚺		ApplicationUnder	Test
8-24	✓ 2	Validate	ContainsImage	Screenshot1		<占 HeaderTop	
-6	🖰 3	Mouse	Click	Left	Relative	😪 OpenDialog	
-156	⊕ 4	Mouse	Click	Left	Relative	<	
i 🗠 🖭 🕯	ергенкероз	nory.rsrep	-			*)	
Ranorex module	browser		🗙 🖸 Add new item 🔹 🍕 Track 🗶 🌐	🖺 📋 🦻 🤗 🕅 🧱 😰 WebTestRepository • 📼 V	/ariables 🎻 Cleanup	- Search (F3)	ρ.
Search (F3)		L L L L L L L L L L L L L L L L L L L	O Item	Path			1 1
😑 👼 WebTest			ApplicationU	InderTest 🕺 EDIT IN SPY	Base: /dom[@domai	n='www.ranorex.com'l	↓ 2
- Groups				and and a section of the section of	and a second		
	ToseBrowser		C. SomePTag	.//div[#/Content]/div[1]/p[3]			
B- Module	s JoseBrowser		C Openualing C SomePTag	.//dw/# Content J/a/@innertext= Open dialog [.//dw/# Content]/dw/1J/p[3]			, _

Wait for action with 10s waiting time. The action waits until the linked repository item exists. This should be the repository item that is manipulated in the following action.

The repository item that the action waits for to exist.

በ Note

2

Your test can still fail if loading takes longer than the time set in the Wait for action and the repository item doesn't exist. Choose a sensible time value that prevents failure because of insignificant loading delays and doesn't keep your test searching for an element for too long.

በ Note

Instead of the Wait for action, you can also use the Delay action to pause your test run.

However, the Delay action always pauses the test run for the specified time, regardless of an element existing or not. This is why you should use the Wait for action if there is a repository item you can sensibly link it to. This way, your test won't idle unnecessarily.

Use the Set value action to enter values more robustly

Entering values like text strings in web forms is a common scenario in web tests. This can either be accomplished by simulating the keypresses on a keyboard (Key sequence action) or by directly setting the form to a specific value. The latter is usually more robust because no mouse clicks or similar are required, so there is less potential for failure. Conversely, this can cause the test to miss certain defects because you deviate farther from a true user experience.

Test situation

In our example, we'll enter a name in the sample form on the Ranorex test website. The process is as follows:



Enter the name in the text field of the form.

Click Submit to submit the name.

The page that appears as a result.

💱 Ranorex Studio - Web test page	× + - • ×
<> < ⇔ < < < < < < < < < < < < < < < < <	https://www.ranorex.com/web-testing-i ••• Q. Search Mr. 🖸 🗶 🚍
Î	Test Form
	Name Harry
	Email 💦 ranorex.com/web-testing-exam 🗙 🕂
	Date $(\rightarrow C)$
	Checkbox
	Color green V Array
٢	Colors green ^ [testname] => Harry 3
	blue [testdate] =>
	red [testcolor] => green
	gray [testmultiple] => Array
	yellow V
	[1] => yellow
	Submit _ 2 _)
)

What it looks like in the action table

💃 WebTest - Ranorex Studio						– 🗆 X
File Edit View Project Build Debug Search	Tools Window N	Help				
Or O 📴 🖬 💾 🖉 🗛 🛤	B 🖬 B	VIEW 🖿 🖄 🗟 🔗 🦁	RUN 🕨 🔳 🏦		🕻 🟥 Default layout 🕤 🔎	:
Projects	■ × WebTest.ntst	Recording1.nrec* InputTed.no	ec* 🗙 InputText2018100			=
== 🖹 C (월 📩 🐸		RD N. RUN			- VA	RIABLES 💽 SETTINGS 🌿 Studio
E-IN Solution WebTest	^					
B References	🔂 Add n	iew action 👻 🗙		9 🤉 🔺 🔻	🛛 🕸 Turbo mode 📑	Screenshot
Reports Ch app.config D Acceptibility or	#	Action	Name	Value		Repository item
B-BM CoseBrowser.txtec B-BM InputText.txtec	01	Mouse	Click	Left	Relative	© Testname
OpenBrowser.nxrec Drogram.cs	898.2	Key sequence	Harry			49 Testname
B-694 Recording 1. norec	🕶 3	Set value	Value	~ Harry	~ 1	⇔ Testname
WebTestratst	04	Mouse	Click	Left	Relative	en Suberit
Saarch(F3) Groups Modules Modules Modules Modules Modules Modules Modules Modules	P Iten B Q He Q Q Q Se C	ationUnderTest esderTop serDislog im#PTap Testname	Path Base: //dom/@idomain='i .//dom/#'header-top] .//dom/#'Content'}/(a)@in .//dom/#'Content'}/(a)@in .//dom/#'Content'}/GDT II	www.tanorec.com] nertext="Open dialog] 83 N SPY	"testname"] 🗧 2	InputTag #tests



Set value action that enters the name in the text field.

Repository item **Testname** that represents the text field.

Result

The value is set directly in the form without any mouse clicks or typing.



For more details on the Set value action, refer to Ranorex Studio fundamentals > Actions > ---> Action properties.

Use the Get value action to read out values for use in the test

It's often useful or necessary to read out values on websites (numbers, strings, etc.) and use them further along in the test. The simplest way to do this is with the **Get value** action.

Test situation

In our example, we'll read out a list value to reuse it in the sample form on the Ranorex test website.

With the default values selected, clicking the **Submit** button produces the following result:





Color/testcolor is set to green.

Colors/testmultiple is set to green and yellow.

We'll now

- get the value blue from the testmultiple parameter and
- set the testcolor parameter to this read-out value.

💥 WebTest - Ranorex Studio						-	0 ×
File Edit View Project Build	Debug Search	Tools Window Help					
0-0 📧 🖬 💾 💣	ADD 📷 😋		🛚 🕅 🗟 🔊 🦁 🛛 RUN 🕨 🛛				
				mec* × WebTest_20101003_100225.cdog			
= 🖿 C 🖭 📩 🐸		e Record	RUN			🖹 SETTINGS	🛵 Studio
Solution WebTest Solution Items WebTest Berences	🔂 Add	new action 👻	K 🗇 🗂 🕯 🛛 🤊	🥲 🛯 🔺 💌 🗍 🖴 Turbo mode	Screenshot		
Reports Pr app.config	#	Action					
Assembly info.cs OnenBrowser.ccrec	₽ 1	Get value	Value	\$Color	1	⇔ Blue	
Program.cs	🗣 2	Set value	TagValue	\$Color	2	♦ Testcolo	r i
B-OH Recording2.nxrec B-OH Recording3.nxrec No WebTest.ntmg	🕀 3	Mouse	Click	Left	Relative	⇔ Submit	\mathbb{T}
Banorex module browner		🛚 🗙 😋 Add new item 🔹 🍕	Tack. 🗶 🗇 🗂 🛔 🤊 🤆 🗐	📰 WebTestRepository + 🖙 Variables 🥑 Cleanup		earch (F3)	p.
Search (F3)	🔂 Add	new item 👻 🔍	Track 🔀 🗇 🗂	📋 🖻 🤆 🔲 🗊 WebTes	stRepository 👻 🖬	Variables	🗹 Cle
Modules Microsoftmuser				Path			
M OpenBrowser	谢 Арр	licationUnderTes	t	Base: /dom[@domain='www.ran	orex.com']		
Recording2	<u>ቀ</u> ‹ አ ዞ	leaderTop		.//div[#'header-top']			
IN- EM Mecorology	···· <2 (OpenDialog		.//div[#'Content']//a[@innertext=	'Open dialog']		
	< ት 5	iomePTag		.//div[#'Content']/div[1]/p[3]			
	<> T	estname		.//input[#'testname']			
	···· 🚸 S	iubmit		.//input[#'submit']			
	····	Blue	🎉 EDIT IN SPY	.//select[#'testmultiple']/option[@	∮value='blue'] 🖛		
		estcolor		.//select[#'testcolor']	+		

What it looks like in the action table

- 1 The Get value action reads out the value blue from the Colors field and passes it to the variable \$Color.
 - The Set value action uses the value of the variable \$Color and sets the field Color/testcolor to it.

Result

2

The resulting page will look like this:

💸 ranorex.com/web-testing-exam 🗙 🕂	
← → C ♠ ① ♣ https://www.ranorex.com/web-testing-exameters	•
<pre>Array ([testname] => [testemail] => [testdate] => [testcolor] => blue [testmultiple] => Array ([01 => present </pre>	
(1) => yellow 2	



The field testcolor now has the value blue, as set by the Set value action.

The testmultiple field is still set to the default values. The Get value action only reads out the value blue.

For more details on the Get value action, refer to Ranorex Studio fundamentals > Actions > ---> Action properties.

WebDocument Adapter

Reference

The WebDocument Adapter creates a representation of the complete website including all tags (e.g. the header tag, body tag, etc.). Furthermore it offers useful ways to make your test scripts more effective.

The following sample shows how to use these features:





Find or filter web elements

The Ranorex Framework offers a wide range of adapters for each HTML tag elements (e.g.: ATag adapter for <a> tags). Each adapter has specific methods and attributes; the link tag (<a>) for example has additional attributes like HREF, TARGET and REL.

C#
// Start IE with a specific website
<pre>System.Diagnostics.Process.Start("iexplore.exe", "/web-testing-examples");</pre>
// Identify the webdocument by its title
<pre>WebDocument webDocument = "/dom[@caption='Ranorex Test Page']";</pre>
<pre>// Find a link by its link text (innertext)</pre>
<pre>ATag link = webDocument.FindSingle(".//a[@innertext='simple link']");</pre>
link.Click();

VB.NET

```
' Start IE with a specific website
System.Diagnostics.Process.Start("iexplore.exe", "/web-testing-examples")
' Identify the webdocument by its title
Dim webDocument As WebDocument = "/dom[@caption='Ranorex Test Page']"
' Find a link by its link text (innertext)
Dim link As ATag = webDocument.FindSingle(".//a[@innertext='simple link']")
link.Click()
```

Repositories and the WebDocument

The following example shows how to access the methods of the WebDocument using a repository:

C#
// Load repository
<pre>ProjectRepository repo = ProjectRepository.Instance;</pre>
// Open a website
<pre>repo.WebPage.Self.Navigate("http://www.ranorex.com");</pre>
// Wait until the document is loaded
<pre>repo.WebPage.Self.WaitForDocumentLoaded();</pre>
VD NET
VD.INE I
VD.INE I
' Load repository
<pre>' Load repository Dim repo As ProjectRepository = ProjectRepository.Instance</pre>
<pre>' Load repository Dim repo As ProjectRepository = ProjectRepository.Instance ' Open a website</pre>
<pre>' Load repository Dim repo As ProjectRepository = ProjectRepository.Instance ' Open a website repo.WebPage.Self.Navigate("http://www.ranorex.com")</pre>
<pre>' Load repository Dim repo As ProjectRepository = ProjectRepository.Instance ' Open a website repo.WebPage.Self.Navigate("http://www.ranorex.com") ' Wait until the document is loaded</pre>
<pre>' Load repository Dim repo As ProjectRepository = ProjectRepository.Instance ' Open a website repo.WebPage.Self.Navigate("http://www.ranorex.com") ' Wait until the document is loaded repo.WebPage.Self.WaitForDocumentLoaded()</pre>

Cross-browser testing

Cross-browser testing involves executing one test across multiple browsers. This can save a lot of time and reduce maintenance efforts, but it comes with its own challenges. In this chapter, we'll go through a cross-browser example step by step.

Test scenario

We want to use Ranorex Studio to test a specific scenario on three different browsers: Microsoft Internet Explorer, Google Chrome, and Mozilla Firefox. We want to accomplish this with only one test that works for all these browsers. To demonstrate our cross-browser example, we'll use a free Dropbox account. **Dropbox** is a registered trademark of Dropbox, Inc. and Dropbox International Unlimited Company. Their terms of services and privacy policy apply. Ranorex GmbH, Ranorex, Inc. and Dropbox, Inc. are not affiliated in any way.



Our test will go through four steps that are defined as follows:

Start

2

Start the browser and go to the URL www.dropbox.com

Click Sign in to reach the sign-in page.



Sign-in

3

Δ

Enter e-mail address and password.

Uncheck the option **Remember me**.

Click **Sign in**.

👽 Login - Dropbox 🛛 🗙 🕂			- 0 X	
← → C* 🏠 🛛 🛈 🖨 Dropbox, Ir	c (US) https://www.dropbox.com/login ***	Q, Search	⊪\© ¥ ≡	
Try Dropbox Business		Download t	he app	
	john48public@gmail.c	om		3
			biast to the	
Je Contraction	Google Privacy Policy and Term	of service.	ibject to the	
	Remember me		Sign in	5

Check account

Here, we'll check if we're signed in to the correct account. There are different ways to do this. In our example, we'll click our account picture and validate the account name.



Click the account picture to open the account menu.

Check if the displayed account name is the same as that of our fictional person (John Public, in our case).



Sign out

8

9



Click Sign out.

Universe - Dropbox		×
Home Files Paper Showcase	Home Btarred When you star items, they Bacant Items you recently viewed	Dohn Public Change photo O bytes of 2 GB used Upgrade
		Settings Install
		Sign out 9

Create a new web test

To get started, first create a new web test as explained in → Build a web test and keep the following in mind:

- Give your solution a meaningful name (e.g. CrossBrowserTest).
- Enter www.dropbox.com when asked for the URL.
- You'll only be able to select one browser in the wizard. This is fine, choose Firefox. We'll extend the test to the other browsers later.
- When asked to select the recording behavior, select Add browsers to whitelist.
- Click **Finish**. The new test solution appears.

Before recording

We'll first record our web test on a single browser. Later, we'll adapt it to work across multiple browsers.

Before we start recording, we'll need to make the following preparations (if you've instructed the wizard to start the browsers automatically, you can skip this).



Start Firefox.

Go to www.dropbox.com

Record the test





Click Sign in.

Enter your e-mail address and password.

- Uncheck Remember me.
- Click Sign in.

Wait until the page has loaded and **click** the account picture.

Validation

Here we'll insert a ----- validation to check whether we're logged in to the right account. We'll do this by validating the account name, John Public in our case.



In the Recorder control panel, **click Validate.**

Ra	nore	x Recorder (Shift	+Esc to stop)				\odot				
8	⊕	Mouse	Left	Click			•				
7	θ	Mouse	Left	Click	Ra	nor	ex Recorder (Shift	+Esc to stop)			\odot
6	θ	Mouse	Left	Click	8	θ	Mouse	Left	Click	0	î
5		Key sequence			7	θ	Mouse	Left	Click	0	î
C		mage based recor	ding 🕕	Enable hotkeys	6	θ	Mouse	Left	Click	0	î
•	R	ECORDING	✓ Validate	Pause S	5	≣	Key sequence			0	î
					C		Image based recor	rding	Enable hotkeys II		
			0		C) v	ALIDATING	☑ Validate	Continue Stop	• A	dd

Mouse over the account name in the account menu. A purple frame will indicate which element is currently selected. **Click** to confirm the selection.





4

2

In the window that opens, **check** if the correct element has been selected. If not, correct the selection. **Click Next** to confirm.

Make sure the attributes Exists and InnerText with the account name are checked. Click OK to confirm.

👯 Home - Dropbox	× +		- 🗆 X
€ → ୯ û	🛈 🖨 Dropbox. Inc (US) https://www.dr	opbox.com/h +++ Q, Search	II\ €D 🗙 🗏
÷	Home	(Q. Search	🟫 Upgrade account
Home Files Paper	Starred When you star items, t	John Public Change photo	2
Showcase	Recent		
	Items you recently view	0 bytes of 2 GB used Upgrade	<

Finish the recording

After the validation is configured, Ranorex Studio will continue recording. It's time to finish the recording.



In the opened account menu, **click Sign out**.

Click Stop in the Recorder control panel to finish the recording.

Result

After stopping the recording, you'll be returned to Ranorex Studio, with the resulting recording being displayed. Let's take a look at this initial version of our cross-browser test.

K CrossBrowserTest - Ranorex Studio							-	\square ×
File Edit View Project Build Debug	Search	Tools Window H	telp					
🕒 🗢 🐹 📅 💾 🖓 ADD	- 🛤 🗬	B 🖬 🗛	VIEW 💼 📆 🛱 🔗 🤨	RUN 🕨 📰 🏦 🛛 BUI		SPY 🔏 🎞	Default layout 👻 🔎	÷
Projects	₽×	CrossBrowserTest.r	ntst Recording1.nrec* 🗙					-
프 🖻 C (엽 📩 🐸		RECORD	NUN				👟 VARIABLES 📑 SETTINGS	1/2 Studio
Solution CrossBrowserTest								
B- R CrossBrowserTest	G Add	new action + 3	K □ □ ■ 9 @ ▲		Contraction Screensh	ot		
References	=	Action						
- Ct app.config	() 1	Mouse	Click		Left	Relative	😪 SignIn	
- AssemblyInfo.cs	🖰 2	Mouse	Click		Left	Relative	LoginEmail77186409088	96506
CloseBrowser.rxrec	▦ 3	Key sequence	john				LoginEmail77186409088	96506
- CrossBrowserTest.rxtst	四4	Key sequence	48public[LControlKey	ublic[LControlKey down][RMenu down]			LoginEmail7718640908896506	
G-E* CrossBrowserTestRepository.r G-B* OpenBrowser.pxrec	05	Mouse	Click		Left	Relative	LoginPassword03784091	2448768
- [] Program.cs	⊞6	Key sequence	E. Subling description	Aller and the			LoginPassword03784091	2448768
Recording1.rxrec	₿7	Mouse	Click		Left	Relative	RememberMe925463212	0187671
	08	Mouse	Click		Left	Relative	⇔ LoginButtonSigninButtor	ButtonPr
	8 9	Mouse	Click		Left	Relative	↔ McPopoverTrigger	
	10	Validate	AttributeEqual		InnerText	John Public	<a>A JohnPublic	
	🖰 11	Mouse	Click		Left	Relative	♀ SignOut	
Ranorex module browser	₽×	😑 📄 Application	UnderTest	Base: /dom]@domain=`www.d	ropbox.com]			
Search (F3)	🔁 Add	new item 🔹 🕵	Track 🛠 🗇 📋 🗎 🗎 🤊	🖓 🖓 📰 CrossB	rowserTestRep	o 🕶 🛏 Variab	oles 🍼 Cleanup 🔠	• Searc 2
Groups	Item			Path				_
CloseBrowser		ApplicationUnde	erTest	Base: /dom[@domain	n='www.dropb	ox.com]		
OpenBrowser		Signin Status Signin	19640008806506	.//div[#"components"	1311484369111	92318']//section/	?/?/ul/?/?/a[@innertext="Sign in"]	
Kecording I		LoginPasswork	rd037840912448768416	.//input/#'login_ena	word03784091	24487684161		
		RememberM	e9254632120187671	.//input[#'remember	me9254632120	187671]		
		LoginButtonS	igninButtonButtonPrimar	.//div[#"component6	2406277032269	92946']//form[@a	action='https://www.dropbox.com	n/login']/div[3],
	8	🗁 PageletTopN	1enuAccountButtonContain	Base: .//div[#'maestro	o-header']/div/	div[2]		
		♦ SignOut		.//nav/?/?/a[@innerte	st='Sign out']			
		S JohnPublic S McPapage	C arTrianar	.//nav/?/?/h2/div/div[1]/div[@innerte	ext='John Public']		
		w wcropov	erngger	:/:/button				



Action table showing 11 actions (your recording may differ slightly in the amount of actions).

Repository with 8 repository items organized into two folders.

Optimize test for cross-browser adaptation

Before we start turning the test into a cross-browser test, we should optimize it.

Combine key sequences

Sometimes Ranorex Studio will split key sequences. This can happen because the sequence wasn't entered fast enough, for example. You should combine key sequences wherever it makes sense.

Select the key sequences you want to combine and open the context menu.

Click Merge selected keyboard items.

🔁 Add i	new action 👻 🗇	📋 📋 🦻 🤗 🔺 💌 🚔 Turbo mode	Sc.	reenshot			
#	Action						
🕀 1	Mouse	Click	Left	R	elative	♀ SignIn	
🖰 2	Mouse	Click		R	elative	LoginEn	nail7718640908896506
≝ 3	Key sequence	john	$\mathbf{P}_{\mathbf{z}}$	Run select	ted item(s)		718640908896506
≕ 4	Key sequence	48public(LControlKey down)(RMenu down)					718640908896506
05	Mouse	Click	0	Add new a	action		prd037840912448768
23 6	Key sequence	E.Suffrag. down(_).Suffrag. op/HDP-able		Enable co	ntinue on fail		prd037840912448768
07	Mouse	Click		Disable		Ctrl+E	e9254632120187671
8 🖰	Mouse	Click	25	Convert to	o user code		SigninButtonButtonPr
09	Mouse	Click	22	Merge ite	ms to user co	de item	rigger
I 0	Validate	AttributeEqual		Move to n	ew recording	module	
🖰 11	Mouse	Click	+C	Merge sel	ected keyboar	rd items	2
		3	Key	sequer	ice	john4	8public{LControlK

Optimize key sequence content

Sometimes, entering special characters can result in unnecessarily complex strings. Simply correct them manually in the action table.

In the example below, we simply replaced the superfluous keypresses with @.

0 ² ⊞ 3	Key sequence	john48public	{LControlK	ey down}{RMenu do	vn}{Qkey}{LControlKey up}{RMenu up}	{LControlKey up}gn	nail.com
⊕ 4	Mouse	Click					
⊞ 5	Key sequence	E.Shifting di	ner(J.Shi	May up/ADPublic			
A ₆	Moure	Click	0 2	Mouse	Click	Left	Relative
			23	Key sequence	john48public@gmail.com	2	
			Δ 🔒	N.4	70.40	1-4	Detexture



Find and replace dynamic identifiers

Many applications contain so-called dynamic UI elements. These changes whenever a particular event happens, e.g. when you reload a web page. It's often harder for automated testing tools to find these UI elements reliably. This is because identifiers that are robust for static UI elements (like the element ID) change all the time for dynamic UI elements. This is why you need to fall back on other identifiers for dynamic UI elements.

For web elements, Ranorex Studio uses an **intelligent algorithm** that recognizes when a UI element is dynamic. It ignores dynamic identifiers and uses a robust, static identifier instead. This means you should normally not need to find and replace dynamic identifiers from your repository items. However, in some cases a dynamic identifier may be missed and you'll have to replace it by hand.

Dynamic identifiers usually appear in the RanoreXPath with a name prefix and a long character string that changes whenever the element is loaded (see paths marked in red in image).



Replacing dynamic identifiers with more robust ones is explained in Ranorex Studio expert > ---> Mapping dynamic UI-Elements.

🔁 Add new item 👻 🍕 Track 🗶 🗇 📋 🔋 🎔	Q ^a □ CrossBrowserTestRepo
Item	Path
ApplicationUnderTest	Base: /dom[@domain='www.dropbox.com']
SignIn	.//div[#'component8131148436911192318'//section/?/?/ul/?/?/a[@innertext='Sign in']
LoginEmail7718640908896506	.//input[#'login_email7718640908896506']
LoginPassword037840912448768416	.//input(#login_password037840912448768416')
 RememberMe9254632120187671 	.//input[#remember_me9254632120187671']
↔ LoginButtonSigninButtonButtonPrimar	.//div[#'component6240627703226992946'//form[@action='https://www.dropbox.com/login']/div[3]
😑 🗁 PageletTopMenuAccountButtonContain	Base: .//div(#'maestro-header')/div/div[2]
<>> SignOut	.//nav/?/?/a[@innertext='Sign out']
< > JohnPublic	.//nav/?/?/h2/div/div[1]/div[@innertext='John Public']
↔ McPopoverTrigger	?/?/button

In our example, the improved repository looks like this:

🚯 Add new item 🔹 🍕 Track 💥 📋 📋 📋 🔊	Q ^e I CrossBrowserTestRepo ▼ ► Variables ✓ Cleanup					
Item	Path					
ApplicationUnderTest	Base: /dom[@domain='www.dropbox.com']					
- 😒 Signin	.//a[@innertext='Sign in']					
↔ LoginEmail	.//div[#'login-or-register-page-content']/div/div/div[1]//form/div[1]//input[@name='login_email']					
<> LoginPassword	LoginPassword .//div[#'login-or-register-page-content']/div/div[1]/div[2]//form/div[1]//input[@name='login_password']					
RememberMe	.//div[#'login-or-register-page-content']/div/div/div[1]/div[2]/div/div/div/div/div/div[3]/div[1]/input[@name='remember_me']					
<> LoginButtonSigninButtonButtonPrimar	.//div[#'login-or-register-page-content']/div/div[1]//form/div[3]/button					
PageletTopMenuAccountButtonContain	Base: .//div[#'maestro-header]/div/div[2]					
<a>SignOut	.//nav/?/?/a[@innertext=`Sign out']					
< JohnPublic	.//nav/?/?/h2/div/div[1]/div[@innertext='John Public]					
<a> OpenAccountInfo	?/?/button//img[@alt='Account photo']					

Empty text fields

In automated testing, it's a good idea to make sure text fields are empty before something is entered in them.

🔂 Add r	new action 👻 🗍	Ё 🗎 🦻 🤉 🔍	≞ ⊪ Turbo m	node 🛛 👩 Scre	enshot
#	Action				
⊕ 1	Mouse	Click	Left	Relative	↔ LoginEmail
拱 2	Key shortcut	Press	Ctrl+A		⇔ LoginEmail
₩3	Key sequence	{Delete}			↔ LoginEmail
₩4	Key sequence	john48public@gmail.com			⇔ LoginEmail
05	Mouse	Click	Left	Relative	⇔ LoginPassword

The four actions in the image represent entering the e-mail address into the text field represented by the repository item **LoginEmail**.

1 2

4

- Click into the text field.
- Press Ctrl + A to select any existing text in the field.
- 3 Press Delete to delete the text.
 - Enter the e-mail address.

If the text field is already empty, actions 2 and 3 won't have any effect. The test will continue without issue. Alternatively, you can also use the Set value action to replace all of these four actions or just the actions for emptying the text field.

Structure your test

Your tests should always be well structured. This is why you should organize your actions into various recording modules and structure them in the test suite.


In our example, we've organized the recorded actions into the following modules and structured the test suite as seen below:

🛠 🗇 📋 🔋 🎔 🭳 🕮 Data source	
Item	Data binding / iterations
☑ ☐ CrossBrowserTest - Test suite	
E TestCase	
ETUP]	
OpenBrowser	
🖸 SignIn	
🖸 Login	D
💽 VerifyAccount	
E- [TEARDOWN]	0
CloseBrowser	



Setup region containing the module for starting the browser and opening the URL.

- Four recording modules to go to the sign in page, sign into Dropbox, check the account, and log out.
- Teardown region containing the module for closing the browser.

Cross-browser functionality

Now that we've optimized the test, we can implement the cross-browser functionality.

Cross-browser testing is based on data-driven testing and variables.



As a first step, we need to replace the fixed browser in our OpenBrowser.rxrec module with a variable. This variable will take on the different browsers as value during the test run.

Open the drop-down menu for the Browser property and click As new variable...

Enter a name for the variable, e.g. selectBrowser, and click OK.

The browser type is now a variable in the actions table.

CrossBro	owserTest.ndst* Open	Browser.nitec 🗙							
	RECORD	เบท							
🗘 Add	new action 🔸 🕅	99	🔺 🐨 🗐 🕸 Turbo mode 🛛	Screenshot					
# 0 1	Action Open browser	URL www.dropbox.com	Browser Firefox IE Firefox Chrome Chromium Edge	Maximized False ~	Add va Variabl	riable e name: t value:	selectBrowser Firefox	2 Cancel	
			As new variable		Add new a Add new a Add new a	Test.ndst* RD ction • 🛠 on en browser	DeenBrowser.narec* ×	v ≞ Turbo mo \$selectBrow	sde lä Screenshot /SET False

Create the data source containing the browsers

In this step, we'll create a data source to provide values to the browser type variable.



Open the context menu of the test case in the test suite view.

Click Data source...

CrossBrowserTest.rxtst 🗙						
ADD V RUN TestRun V Maintenance mode						
🛠 🗇 🖺 🛢 🛛 🤊 🖓 📲 Data source						
Item	Data binding / iterations		I	Description		
CrossBrowserTest - Test suite						
E SETUP	Inhound variable: 1	, ₽	Run selected test c Run to here (excl. s	ase selection)		
SignIn	onboaria variablei r	T	Data source	2		
O∢ Login O∢ VerifyAccount		œ	Data binding Condition			
Get SignOut (TEARDOWN) Get CloseBrowser		0	Add Add setup Add teardown	•		

Click New > Simple data table and **enter** a name for the table, e.g. **BrowserList**.

Use the Add column... and Add row... buttons to create the table and fill it with the values in the image below.

Click OK.

3

4

5

TestCase properties	— 🗆 X
TestCase	
General Data source Data binding Condition	
New 3 BrowserList - SimpleDataConnector ~	Manage data sources
Browser Add column 1 Chrome 2 Firefox 3 IE 4 Add row	
All rows Range e.g. 1-5, 8, 11-13	Refresh Preview effective data set
J OK	Cancer Apply



Creating data sources is explained in

Ranorex Studio advanced > Data-driven testing > ---> Data and data management.

Binding the data to the variable

Open the context menu of the test case in the test suite view.

Click Data binding...

CrossBrowserTest.rxtst* ×							
🕒 ADD 🗸 🕨 RUN TestRun	D V Naintenance mode						
🛠 🗇 📋 🛢 🛛 🤊 🖓 🎞 Data source			Searc				
Item	Data binding / iterations		Description				
CrossBrowserTest - Test suite							
= 🗹 💽 TestCase	BrowserList 🛢 Rows: 3 📋		Run calected test case				
		E.	Run to here (excl_selection)				
OpenBrowser	Unbound variable: 1						
SignIn	ſ		Data source				
Of Login			Data binding 2				
VerifyAccount		G	Condition				
SignOut		0	Add +				
E. [TEARDOWN]			Add setup				
CloseBrowser			Add teardown				

3

4

Under Variable binding > Module variable, select the variable OpenBrowser.selectBrowser from the drop-down menu to bind it to the Browser column of the data source.

Click OK.

Test	~					
	Case					
neral Da	ata source Data binding	Condition				
Variable b	pinding					
Data col	lumn	Module	variable			
Browser		OpenBrov	vser.selectBrowser		\sim	3
		Open B	rowser.selectBrowser		-11	
			Auto-bind C	lear binding	js	
Paramete	ers			In	fo -	
	Name	Value	Module variab	le		
▶ 1	Add row			1	~	
		Auto-cre	ate Auto-bind C	lear binding	IS	
		4	OK Cancel	Appl	ly	
Ľ	h					
	Reference					
Dat	a binding is des	cribed in Rand	orex Studio adv	vanced	> Da	ata-driven testing >> 🛙
bing	ding.					

Run the cross-browser test

The final cross-browser tests should look as follows in the test suite:

CrossBrowserTest.nxtst 🗙								
G ADD V NUN	TestRun	✓ Maintenance mode						
🛠 🗇 🖆 🔋 🦻 🥲 🕮 Data source								
Item	1	Data binding / iterations						
☑ 📄 CrossBrowserTest - Test suite								
🖃 🔽 💽 TestCase		BrowserList Rows: 3						
EETUP]								
OpenBrowser		Bound variable: 1						
🔤 🖸 SignIn	1							
🖸 Login								
🖸 VerifyAccount								
🖸 SignOut								
E. [TEARDOWN]								
CloseBrowser								
	4							



- The test case contains a data source called **BrowserList** with 3 rows of data.
- The module **OpenBrowser** contains one variable that is bound to a data source.

Results

2

3

After the test run, the report should look as follows:

x Ranorex Studio		
CrossBrowserTest		
Computer/Endpoint	Execution time 10/10/2018 8:22:31 AM	Test case result summary
Operating system Windows 10 64bit	Screen dimensions 5760x1200	
OS Language en-US	Duration 2m	3x Success
Total errors 0	Total warnings 0	
Expand test containers Expand	details Collapse all	
Test container filter: 🗹 Success	Failed Blocked	
A TartCare Pour 2	2	



Three test cases were completed successfully.

Our single test case was iterated three times, once for each row in the data source, i.e. once for each browser. Hence, three successful test cases.

Bringing up the details of the **OpenBrowser** module for an iteration shows the value passed to the variable:

▼ 📑 TestCase Rows: 3			2m	
TestCase Data Row: 1			33.54s	
* Test data Browsen: Chrome				
▼ SETUP			899ms	
- 🔀 OpenBrowser			873ms	
Filter: 🗹 Info				
Time Level Category Message				
00:01.217 Info Data	Current variable values: \$selectBrowser = 'Chrome'			
00:01.352 Info Website	Opening web site 'www.dropb	oox.com' with browser	specified by variable \$select	Browser in normal mode .
· De Verileitoreunt			8.51	
Sign Dut			1.95z	
TEARDOWN			112ms	



Reports are explained in Ranorex Studio fundamentals > ---> Reporting.

Download the sample solution

You can download the completed sample solution file from the link below:

Sample Cross Browser Test

Attention

The recording module Login.rxrec **does not contain any login data**. **Fill in your own** Dropbox account data. You will also need to **change** the content of the **Validation attribute InnerText** to your own account name in the recording module VerifyAccount.rxrec.

Install the sample solution:



Δ

Unzip to any folder on your computer.

Start Ranorex Studio and open the solution file CrossBrowserTest.rxsln

O Hint

The sample solution is available for Ranorex versions 8.3 or higher. You must agree to the automatic solution upgrade for later versions.

Endpoints

Endpoints are the gateways through which a locally executed test exchanges data with an external AUT. In simpler words, they allow you to test an external application or system as if it were on your machine.

Manage endpoints

Endpoints that have been added to Ranorex Studio are displayed and managed in the Endpoints pad. Click the endpoints symbol in the menu bar to bring up the Endpoints pad. By default, it appears on the right edge of the screen.





Empty Endpoints pad.

Endpoints pad displaying the endpoint list with three endpoints.

Endpoint list

If your endpoint list is empty, you'll see a corresponding message and a button to add an endpoint. If you've already added at least one endpoint, the endpoint list will appear as follows:



- Refresh and Add endpoint buttons.
- 2 Search field for endpoints.
- 3 Click to access this endpoint's settings.
- 4 Endpoint name
- 5 Connection status (Connected/Error/Connecting...)
- 6 Endpoint ID: IP number for network connections, ID for USB connections.
- 7 Automation root status symbol (explained in → Endpoint settings)
- 8 Endpoint type (Android, iOS, WebDriver)

Shortcuts and command line execution

Aside from the usual test executable, the /output/ folder also contains shortcuts for each endpoint. These shortcuts allow you to execute the test directly on a specific endpoint. The endpoint is represented in the shortcut's file name as @<endpoint>.

	AndroidAppTest.exe		Application	20 KB			
	AndroidAppTest.exe.	config	XML Configuratio	1 KB			
	AndroidAppTest.pdb		PDB File	32 KB			
	🔄 AndroidAppTest.rxtmg		Ranorex Test Suite	1 KB			
_	AndroidAppTest.rxtst		Ranorex Test Suite	3 KB			
	🗾 AndroidAppTest@US	B-00c18ccf5e6b3703	Shortcut	2 KB	1 Duration File		
	Ranorex.Core.Resolve	er.dll	Application extens	206 KB	ile		
		iooAppresting.pub		TOOTI	lie		
	🖳 iOSAppTesting.rxtmg			Ranor	ex Test Suite		
	iOSAppTesting.nxtst iOSAppTesting@RX iPad		Ranorex Test Suite				
				Shorto	cut	6	2
		Ranorex.Core.Resolver.dll		Applic	ation extension		



Shortcut for an Android endpoint.

Shortcut for an iOS endpoint.

Reference

You can also manually trigger test runs on endpoints through the command line. The required arguments are listed in Ranorex Studio fundamentals > Test suite > ---> Run tests without Ranorex Studio.

Endpoint settings

In this chapter, we'll explain the available settings for endpoints.



Most of the settings explained here are valid for Android, iOS, and WebDriver endpoints. WebDriver endpoints have some additional, more complex settings that we explain separately in ---> Add a WebDriver endpoint.

Endpoint settings

Click the button to the right of an endpoint to bring up its settings.



Set as automation root

- Click to set an endpoint as the automation root.
- This means that tests will only be able to interact with this endpoint and will receive and send all data for test execution through this endpoint.
- To reflect this, the run buttons in the Ranorex Studio toolbar and the test suite/recorder view will change accordingly. You can also set an endpoint as the automation root by clicking the symbol to its left in the endpoint list.



- **View details –** Displays a detailed view of the endpoint's properties:
- Connection status and type

2

- Endpoint name
- Ranorex Service (version of the Ranorex Service App)
- OS of the mobile device (Android/iOS only)
- Address
- Instrumented apps (Android/iOS only. Click the symbol next to an app to start it on the device. Click the rocket button to → instrument and deploy an app to the device).

3 Instrument and deploy app (Android/iOS only)

- Click to ---> instrument and deploy an app to the mobile device.
- Opens the ---> Ranorex Instrumentation Wizard.



Save ADB log (Android only)

• Saves the log file of the Android debug bridge.

Refresh/Edit/Delete – self-explanatory

🛕 Attention

Be careful when editing the endpoint's address. A wrong address will result in a connection error and test failure.

Add an Android/iOS endpoint

In this chapter, you'll learn how to add an Android or iOS endpoint.

---> Adding a WebDriver endpoint is explained separately.

Preparation





Turn on your mobile device and connect it to a power source.

Start the Ranorex Service App.

Connect the device to your computer via USB (recommended) or WiFi (same network as computer).

Add endpoint

3





An empty endpoint list.

Endpoint list with Localhost (the computer) and two endpoints added.



In an empty endpoint list, click Add endpoint. Or, in a list of existing endpoints, click +.



Select the OS of your mobile device and then **select** your device from the list below. You can also enter the device's address manually by clicking **Other device... Click Add endpoint**.

3

4



በ Note

If your device doesn't appear in the list, refresh it and make sure all required \rightarrow device settings have been applied.



Click Complete setup to return to the endpoint list or **Add another endpoint**. In either case, the newly-added endpoint appears in the endpoint list.



Add a WebDriver endpoint

In this chapter, you'll learn how to add and configure WebDriver endpoints.

Preparation

Set up a WebDriver infrastructure, start the server, and note its address.

Ranorex Studio supports Selenium and Appium through WebDriver endpoints.

Selenium WebDriver

• Setting up a Selenium infrastructure is covered in --> Selenium WebDriver integration.

Appium WebDriver

• On Appium, Ranorex Studio currently supports only mobile web tests.

- For help with setting up an Appium infrastructure, please refer to the official Appium documentation.
- Also, make sure you set the **command timeout to at least 30 seconds** in the → plugin settings for WebDriver.

Add an endpoint for WebDriver





2

An empty endpoint list.

Endpoint list with Localhost (the computer) and two additional endpoints.

In an empty endpoint list, **click Add endpoint**. Or, in list that is not empty, **click +**.





Select WebDriver.

Enter a name for this endpoint.

Enter the server address.

በ Note

In a default Selenium environment, you must add /wd/hub to the URL or IP address of your Selenium server. Only change or omit this path if you have set up your Selenium infrastructure differently.



6 7 **Click Test connection** to see if a connection can be established. If the test fails, **check** that your server address is correct and that the server is running. **Click Add endpoint** and **Complete setup**.

http://localhost:4444/wd/hub Example http://localhost:4444/wd/hub				
Connected	Test connection	6		
	Add endpoint	0		
	④ Comp	olete setup		
			Endpoint successfully addec	i
			Add another endpoint 🗕	Complete setup



The new WebDriver endpoint appears in the endpoint list.



Add an endpoint configuration to a WebDriver endpoint

Endpoint configurations allow you to tell a Selenium WebDriver endpoint how to behave during test execution. Selenium calls these characteristics of an endpoint capabilities. Most commonly, they define what OS, browser, and browser version the WebDriver endpoint simulates. In Ranorex, the sum of all capabilities defined for an endpoint is called an endpoint configuration.

Endpoint configurations make it possible to easily execute tests in a Selenium grid on multiple different system configurations without actually needing to have any of these systems present.

To add a configuration to an endpoint:

Click the symbol to the right of a Selenium WebDriver endpoint.

Click View details.



Click Add new configuration.

3

Endpoints	₽ ×	
Back MySelenium		
Endpoint information:		
Status: Connected		1
Endpoint name: MySelenium	•	-
Address: http://localhost:4444/wd/hu	Ь	
Currently running browsers		
No browsers are currently r	unning.	2]
~	Open URL in 🚽	
Endpoint configurations	Add new configuration	B
Active configuration:	~	
Available configurations		
No configurations add	ed	

1 ^E 2 ^L

Endpoint information showing name, connection status, and server address.

List of currently running browsers with browser name and URL.

Specify the endpoint configuration



3

Enter a name.

Enter a description (optional).

Enter the configuration, i.e. the capabilities definition, as JSON. Make sure it conforms to the requirements of your Selenium Grid provider. Selenium provides a list of capabilities. In the screenshot, the capabilities JSON conforms to the requirements for Sauce Labs.



Note

0

If you are unfamiliar with JSON, use the Ranorex Studio \rightarrow capabilities configurator. Simply select your capabilities and provider and let the configurator put together the JSON. Then, just paste the JSON into the Capabilities JSON box of the endpoint.



Click Save to confirm. The configuration appears in the endpoint details.



Capabilities configurator

The capabilities configurator lets you generate your capabilities as a JSON in an easy and straightforward way.

Simply select your desired options in the fields below. If you want to connect to a service like SauceLabs, select the appropriate option and then replace the placeholders in the JSON with your credentials.

Finally, copy the generated JSON and paste it into the 'Capabilities JSON' text box of your configuration in Ranorex Studio.

Platform			
	Ś	۵	6

Linux

Windows Mac

Appium/Mobile

Select the platform you want your test to run on. Currently, PC and Mac systems are supported.

Operating System	
Select Operating System	*
Browser	
Select Browser	~
Browser Version	
Select Browser Version	~

Provider

Select Provider Some providers require specially formatted credentials for logins. Replace the placeholder with your actual data when pasting the JSON into your application.



Ranorex Parallel Runner

Ranorex Parallel Runner is a command-line tool that allows you to run a test across multiple capability sets on a Selenium Grid through a WebDriver endpoint in parallel. The tool reads the capability sets from an Excel template file.

Using Ranorex Parallel Runner

This tool is located in the **Bin** folder of your Ranorex installation. By default, this is C:\Program Files (x86)\Ranorex\Bin\Ranorex.ParallelRunner.exe

It's a good idea to run the tool with **-help** for the first time to display all available arguments:

```
Command Prompt
                                                                                                       \times
C:\Program Files (x86)\Ranorex 8.0\Bin≻Ranorex.ParallelRunner.exe --help
Ranorex Parallel Runner
opyright © 2017 Ranorex GmbH
                                                                                                               2
Arguments:
 -t, --testsuite
                      The .exe file of the test suite to be executed in parallel for the given
                      capability sets.
                      Excel file containing the capability sets.
 -m, --matrix
 -h, --help
                      Displays this command list.
 -e, --endpoint
                      Name of the endpoint where the test suite is being executed.
                      Prints all capability combinations from the Excel file to the console.
In combination with the print command, the capabilities are displayed in JSON
 -j, --json
                      format.
 -o, --output
                      Relative or absolute output directory for JSON config and report files.
 -c, --create
                      Creates a template Excel for defining capability sets.
Jsage:
  Ranorex.ParallelRunner.exe -t <file> -m <file> -e <endpoint> [-o <path>]
  Ranorex.ParallelRunner.exe -c [<file>]
  Ranorex.ParallelRunner.exe -m <file> -p [-j]
:\Program Files (x86)\Ranorex 8.0\Bin>
```

Under the **Usage:** header, the three different use cases for the tool and the required arguments are listed.

```
Ranorex.ParallelRunner.exe -c []
```

To get started, you'll need to specify a set of capabilities. To do so, use this argument to create the required Excel template. You can specify a location and filename if desired. By default, the file will be created in the **Bin** folder and called **CapabilitiesMatrix.xlsx**. Follow the instructions in the Excel file and define your capabilities.

Ranorex.ParallelRunner.exe -t -m -e [-o]

Use this argument pattern to run your test with the defined capability sets on the specified WebDriver endpoint in parallel. The –o argument is optional. If you don't specify a path, the Parallel Runner will generate a folder called **parallelRuns** in the folder where the test suite executable is located.

```
Ranorex.ParallelRunner.exe -m -p [-j]
```

Mobile Testing

In this chapter, you'll learn how to test apps and websites on mobile devices including smartphones and tablets. You'll learn how to connect devices to Ranorex Studio, instrument apps, and create and run automated mobile tests.

System requirements

Supported devices

Ranorex Studio supports Android and iOS mobile devices.

Connection

Mobile testing requires that you connect your device to the computer that Ranorex Studio is installed on. You can do so by USB (recommended) or WiFi.

Note

0

Ranorex iOS implementation requires that ports 31000 TCP and 31000 UDP must be open on your computer.

Additionally, Ranorex Android requires other ports above 31000 TCP; therefore, enable ports up to 31020.

On some Android devices, such as Huawei, ports 31000 – 31900 are already used. However, the RxService application chooses the next available ports, which are detailed in a message after the RxServices installation. Make sure to open these ports on your computer and correctly configured the Endpoints in Ranorex Studio for Discovery and Device port fields.



Use the correct automation libraries for mobile testing

Test automation of Android/iOS apps is based on the Ranorex automation libraries for mobile testing. To ensure automation works smoothly, always use the automation libraries that correspond to your currently installed version of Ranorex Studio. Otherwise, you will receive technology limitation warnings or even errors.

You can download the automation libraries for the different Ranorex Studio versions from the mobile download archive.

If you absolutely must use earlier automation libraries, you can do so as long as they only produce warnings, but this is not recommended. In this case, simply close the technology limitation warning.

The basic procedure of mobile testing

Mobile tests in Ranorex Studio always follow the same basic procedure, regardless of the type of device. The mobile OS and the type of test (app test or mobile web test) then determine the specific actions and settings in the steps this basic procedure entails.



Prepare your device

- Enter the recommended device settings for Android and iOS
- Install the Ranorex Service App

Connect your device

- Choose a connection type
- Add the device as an endpoint in Ranorex Studio

Instrument apps

- Pre-instrumented Ranorex web browser (Android, iOS)
- Instrument Android apps
- Instrument iOS apps
- Pre-instrumented iOS sample app

Create a mobile test

- Create a solution for mobile testing
- Choose technology and test type
- Create the mobile test

Run a mobile test and get the report

The following chapters explain each of these steps in detail.

Prepare your device

Before you can connect your device to Ranorex Studio and add it as an endpoint, you need to make certain preparations.

Android device settings

Activate developer mode

Mobile testing requires that some developer mode settings be activated. This mode is normally hidden on Android devices. To activate it:



Go to your device's settings.

Find the **Build number** entry. It's normally located in the **About** section, but the exact location may vary depending on device manufacturer and Android version. Refer to your device's documentation for more information.



Tap the Build number quickly seven times.

The notification **You are now a developer** indicates that developer mode is now active.

1	The settings should now o	contaiı	n an	entr	y call	ea D	eve	lope	er op	tion
1	«[]» 🖤 🖹 🗎 9:45									
← Sys	stem									
Gbo	nguages & input									
	Backup									
G	On									
{}	Developer options	1								
	System update									
Ŧ	Updated to Android 8.0.0									
⊲ ctiva	ate Stav awake									
⊲ activa ±∎ ←	ate Stay awake Developer options	10:09 ?								
⊲ ctiva ± ■ ←	ate Stay awake Developer options On	10:09?								
⊲ ctiva ≮	ate Stay awake Developer options On Memory	10:09?								
⊲ ctiva ←	A C C C C C C C C C C C C C C C C C C C	 10:09 ? 								
⊲ ctiva ≮	A CONSTRUCTIONS	iii 10:09 ? •								
⊲ activa ←	Ate Stay awake	iii 10:09 2								
⊲ ctiva ± ■ ←	A CONCEPTIONS Developer options On Memory Take bug report Desktop backup password Desktop full backups aren't currently protect tay awake creen will never sleep while	10:09 ? eed	ng							

Enable USB debugging



In the Developer options, enable USB debugging.

Confirm.



🛕 Attention

For some Android devices, USB debugging does not work with the default Windows USB driver. This makes testing with Ranorex Studio impossible. In these cases, you need to install the device manufacturer's **special developer USB drivers**.

You can normally see that this is the case if Windows recognizes your device correctly, but you can't add it as an endpoint in Ranorex Studio.

Connect your device to a power source

Regardless of the connection type (USB/WiFi), make sure that your device is connected to a power source at all times during test creation and execution.

This ensures that everything runs as intended and is particularly important for WiFi connections, as it deactivates power saving.

iOS device settings

Deactivate Auto-Lock



Go to your device's settings.



Connect your device to a power source

Regardless of the connection type (USB/WiFi), make sure that your device is connected to a power source at all times during test creation and execution.

This ensures that everything runs as intended and is particularly important for WiFi connections, as it deactivates power saving.

Install the Ranorex Service App

To create and run mobile tests on your mobile device, you need to install the Ranorex Service App on your device.

You can install it on your device by running the Instrumentation Wizard or by downloading the app.

Update Service Instrumentation Wizard

After you have added your Android device as an Endpoint, run the Instrumentation Wizard.

1. Select Instrument a mobile device and click Next.



2. Choose between iOS and Android then click Next

*	
• •	•
Instrument a Choose yo	mobile device
iOS	Android
Prepare an iOS device for mobile tests	Prepare an Android device for mobile tests
Create mobile tests in Ranorex Studio	< Back Next >

Х

3. Under the Instrument a mobile device screen, select a device from the drop-down list, and **click Next**.

X Ranorex Instrumentation Wizard

	*			?
	• •			
	Instrument a Choose the device ye	mobile device		
	Choose from existing devices	iPh	ion v	
		or		
	Add a new device Use the Endpoints pad to add ne devices.	w Endpoint	5	
Instrument iOS devices i	n Ranorex Studio		< Back	Next >

Х

4. When the Ranorex Service is not installed on your device, the Instrumentation Wizard opens the **Update Service** tab. **Select** the **Update checkboxes** for **Ranorex Service** and **Ranorex Web Testing** and **click Finish**.





 \times

5. Once the **Operation successful** message displays, **click Finish**.





Download Ranorex Service App

Scan the QR code below or go to https://www.ranorex.com/rxApp to download the Ranorex Service App.



Mobile download archive

The mobile download archive is categorized according to Ranorex Studio versions and mobile OS. Simply download the correct Ranorex Service App.

 \times

Mobile Download Archive

This page provides download links for the latest Ranorex automation libs and service apps.

Note: A debug build with a debug profile is required to start and stop iOS apps via USB. The directly-installable IPA packages below have a distribution profile, and do not support starting and stopping apps via USB. iOS distributables valid until November 11, 2021.

~	Ranorex 9.5.1 - 10.0.0 Android (RxLibVersion 2.3.3)
	Service App: Service App 2.3.3
	Android Web Testing: RxBrowser 2.3.3
	IOS Directly Installable on Device
	Service App: RxServiceApp Distributable 3.2.4 iOS Web Testing: RxBrowser Distributable 3.2.4 Sample App: RxMiniKeePass Distributable 3.2.4
	Manual Deployment RxAutomation Library: libRxAutomationUni 3.2.4
1	Ranorex Service App for Android
2	Ranorex Service App for iOS
Install	the Ranorex Service App on Android Tap the download link.
2	Confirm the security dialog.
Ar	ndroid (RxLibVersion 2.3.3)
•	Service App Service App 2.3.3
•	Android Web Testing: RxBrowser 2.3.3



Tap Settings in the following dialog.

Enable Allow from this source.


Confirm the access dialog.

The Ranorex Service App has been installed successfully.



Your Android device is now ready to be ---- connected to Ranorex Studio.

Install the Ranorex Service App on iOS

Tap the download link.



5

Confirm the security dialog.



The Ranorex Service App appears on the home screen.

3

Δ

iOS considers the Ranorex Service App "untrusted" because it wasn't downloaded from the App Store. **Close** the dialog.



Confirm Ranorex Service App as trusted

Before you can use the Service App, it needs to be trusted.



Go to **Settings > General**.

Go to Device Management.

ad 🔻	09:51		\$ 81 % 🔳)	
	< General	Device Management		
Settings	ENTERPRISE APP			
Q, Search	Ranorex GmbH		3 >	
C Andreas Aussenficher Aussell, Daub Versell Australia		Apps from developer "Phone Dis run until the developer is trusted	stribution: Ranorex GmbH" are not trusted	on this iPad and will not
≻ Airplane Mode	L			
🛜 Wi-Fi A1-11d041		APPS FROM DEVELOPER "IPHOP	NE DISTRIBUTION: RANOREX GMBH*	Trust "iPhone Distribution: Ranorex GmbH" Apps on This
Bluetooth On		RxServiceApp		iPad Trusting will allow any app from this enterprise developer to be used on
Notifications				your iPad and may allow access to your data.
Control Centre				Cancel 5 Trust
C Do Not Disturb				
				Delete App
O General				
A Display & Brightness			APPS FROM DEVELOPER "IPHONE DIS	tribution: ranorex gmbh* Verified



Your iOS device is now ready to be connected to Ranorex Studio.

Connect mobile devices

Before you can create and run tests on your mobile device, you need to connect it to your computer and to Ranorex Studio. This requires adding your device as an endpoint in Ranorex Studio.

Connection types

You can connect your device to the computer that Ranorex Studio is installed on via USB or WiFi. The following tables illustrate the differences between these connection types for Android and iOS. **We recommend using a USB connection**, as this has no restrictions in terms of functionality and is generally more reliable.



Ports 31000 TCP and 31000 UDP must be open on your computer.

	Android		iC	S	
	USB	Wi-Fi	USB	Wi-Fi	
Create tests	\bigcirc	\bigcirc	\bigcirc	\bigcirc	
Run tests	\bigcirc	\bigcirc	\bigcirc	\bigcirc	
Start/stop app	\bigcirc	\bigcirc	\bigcirc	\bigcirc	
Install APK	\bigcirc	\bigcirc			
Install IPA			\bigcirc	\otimes	
Deploy (unattended)	\bigcirc	\otimes	\bigcirc	\otimes	
Automate system apps	\bigcirc	\otimes			

Connection requirements

Android USB connection

For some Android devices, USB debugging does not work with the default Windows USB driver. This makes testing with Ranorex Studio impossible. In these cases, you need to install the device manufacturer's **special developer USB drivers**.

You can normally see that this is the case if Windows recognizes your devices correctly, but you can't add it as an endpoint in Ranorex Studio.

iOS USB connection

You must install iTunes to connect your iOS device to your computer via USB. It provides the required drivers.

General network requirements and recommendations Connect device to a power source

Regardless of the connection type (USB/WiFi), make sure that your device is connected to a power source at all times during test creation and execution.

WiFi network

When connecting via WiFi, the mobile device must be in the same network as the computer.

USB hubs

When connecting via USB, avoid USB hubs. Always use your computers integrated USB ports if possible.

Connect a device and add it as an endpoint



In this example, the device is connected via USB. The process is basically the same when connected via WiFi.

Connect the device to your computer via USB

For Android devices, a short sound will normally indicate that the device has been recognized.

For iOS devices, iTunes will automatically open if the device is recognized.

The device also appears in your Windows settings if connected properly.

Add the device as an endpoint in Ranorex Studio



Adding a mobile device as an endpoint is explained in

Web and mobile testing > Endpoints > ---> Add an Android/iOS endpoint

Instrument apps

Before you can use test automation with a mobile app, you must instrument the app. Instrumenting an app means configuring the app so that Ranorex Studio can access the GUI and functions of the app for the purpose of creating and running an automated test.

Instrumenting apps works differently for Android and iOS, which is why the procedures are explained separately in this chapter.



Attention

Whenever you update Ranorex Studio (i.e. use newer automation libraries/Ranorex Service App), **reinstrument your apps**. Otherwise, automation may not work as intended.

Attention

For iOS developers: As part of the instrumentation process, the app will be compiled with the Ranorex automation library. This library adds additional functions and permissions to your IPA.

Do NOT publish instrumented apps to the App Store.

Instrumented apps **do not work** with TestFlight.

Preinstrumented Ranorex Web Browser for Android

For web tests on Android devices, the preinstrumented Ranorex web browser for Android is required.

Download and install the Ranorex Web Browser



Scan the QR code below or **go** to https://www.ranorex.com/rxApp to download the Ranorex Web Browser for Android.



Select the preinstrumented browser in the mobile download archive:

Μ	obile Download Archive
This stop	page provides download links for the latest Ranorex automation libs and service apps. Note: A debug build with a debug profile is required to start and iOS apps via USB. The directly-installable IPA packages below have a distribution profile, and do not support starting and stopping apps via USB.
~	Ranorex 8.2.0 Android (RxLibVersion 2.0.1) • Service App: Service App 2.0.1 • Android Web Testing: RxBrowser 2.0.1



Install it as you would any other app on your mobile device.

	× ⊻ ⊜ ⊡	4 4	💎 🖹 🛿 10:44	MEXIOUPP V	1 0:44
Enter v	veb address	s here		🐥 Ranorex Services	:
BACK	FORWARD	REFRESH	GO	Nexus 5X Model: Nexus 5X Service Version: 2.0.1	
				AutomationLib: 2.0.1	
				Edit	
	\bigtriangledown	0			
		1		2	



Opened Ranorex Web Browser.

The app is also listed in the Ranorex Service App, which indicates that it has been instrumented correctly.

Preinstrumented Ranorex Web Browser for iOS

For web tests on iOS devices, the preinstrumented Ranorex web browser for iOS is required.

Download and install the Ranorex web browser



Scan the QR code below or **go** to https://www.ranorex.com/rxApp to download the Ranorex Web Browser for iOS.





Select the preinstrumented browser in the mobile download archive:

3 Install it as you would any other app on your mobile device.

6-93 W	11-53	6 64 W	11-53	4 84 14
ReBrowser v3.0.2: Enter web address here		100	Deneroy Corviese Anne	
		00	Ranorex Services - Apps	6
			RxBrowser	
			References and a second	
			hangounded tool c	
500				
1 3 3 4 5 q w e r t	° 7 ° ° y u i o	p 🗷		
a s d f	g h j k l	return		
z x c v	bnm '			
.?123 🛞 : /	com .?13	23 🔤	i kha 🗮 kana	D in
	-			
	1		2	

1	
2	

Opened Ranorex Web Browser.

The app is also listed in the Ranorex Service App, which indicates that it has been instrumented correctly.

Instrument an Android app

You need to instrument Android apps before you can use them for mobile tests with Ranorex Studio.

Note

0

It's not possible to instrument apps that have been downloaded from Google Play.



Instrumenting Android apps is explained in the following chapter:

Interfaces and connectivity > Ranorex Instrumentation Wizard > ---> Android apps.

Instrument an iOS app

You need to instrument iOS apps before you can use them for mobile tests with Ranorex Studio.

🛕 Attention

Instrumenting iOS apps is challenging and not self-explanatory. It's best done by **experienced iOS developers**. To allow you to evaluate iOS testing with Ranorex Studio without the hassle of instrumenting your own app, we provide the **preinstrumented KeePass app** for your use.

Attention

For iOS developers: As part of the instrumentation process, the app will be compiled with the Ranorex automation library. This library adds additional functions and permissions to your IPA.

Do NOT publish instrumented apps to the App Store.

Instrumented apps do not work with TestFlight.

Note

It's not possible to instrument apps that have been downloaded from the App Store, as they are subject to DRM restrictions.

Instrumenting an iOS app means integrating a special Ranorex library (Ranorex libs) into the source code of the app and then compiling it. Afterwards, the app needs to be signed with a P12 certificate and the proper provisioning profile.

For more information on iOS development and release procedures, please refer to the official Apple documentation.

There are two ways to instrument iOS apps:

• With the Ranorex Instrumentation Wizard (recommended)

-	
	_
_	_
	_

Reference

Instrumenting iOS apps using the Instrumentation Wizard is explained in:

Interfaces and connectivity > Ranorex Instrumentation Wizard > ---- iOS apps

 Manual instrumentation of the source code in iOS with Xcode iOS developers can integrate the Ranorex library, compile it, and sign the app manually using Xcode.



Reference

Manual instrumentation is explained in the following chapter:

Web and mobile testing > Advanced mobile testing > \rightarrow iOS source-code instrumentation



After instrumenting and deploying the iOS app for the first time, **start the app manually once!** Otherwise, it will not be visible within the Ranorex Service App.

Preinstrumented iOS sample app

Ranorex provides a preinstrumented iOS sample app for evaluation and learning purposes. The app is **KeyPass for iOS**, distributed under the GNU General Publice License. For information regarding the copyright holder and the license agreement, go to https://keepass.info/help/v1/license.html.

Download and install the sample app



Scan the QR code below or go to https://www.ranorex.com/rxApp.





3

The mobile download archive is categorized according to Ranorex Studio versions and mobile OS. **Download** the RxMiniKeePass app for iOS devices for your Ranorex Studio version.

Next, **install** the sample app as you would any other app on your device.





The RxMiniKeypass app appears on your home screen.

Since the app is already instrumented, it also automatically appears in the Ranorex Service App.

Create a mobile test

In this chapter, you'll learn about the basics of creating a mobile test in Ranorex Studio.

Prerequisites

Before creating a mobile test, make sure you've completed all the necessary preparations:



2

2

Apply the necessary → device settings to your device and **install** the Ranorex Service App.

----> Connect your device to the computer and ---> add it as an endpoint in Ranorex Studio.

3 Finally:

а

b

If you want to test a mobile app, \rightarrow instrument the app.

If you want to carry out a web test on your device, → install the preinstrumented Ranorex Web Browser for Android (recommended) and/or iOS (required).

Create a solution for mobile testing

To create a mobile test, you must use the RocketStart Wizard.

On the Ranorex Studio start page, click New solution using wizard... or click File > New > New solution using wizard...

Click Mobile.



Follow the instructions of the wizard.

When you get to the screen below, choose which kind of mobile test you want to create and click Continue.

- Mobile iOS app test а
- Mobile Android app test b

3

Δ

- Mobile iOS web test
- Mobile Android web test d



Depending on your choice, the wizard will show you a checklist of necessary preparations. If you followed our instructions under Prerequisites, you should already have completed all of them. **Click Continue**.



The prebuilt web test will look like this:

5

🗶 Maintellusi - Karaona Biantu		-	в ×			
All for the Paper field bong back bon O O O D D D D D O O O O O O O O O O O O	MobileTest.rxtst × 1	☐ I Detroit tepost. → Ø I Tepracy in herein (notion Mathematication) - Convers Studie	•			- 8 X
Declare Materials Declare Materials	🕒 ADD 🗸 🕨 🕨	In the per part and party parts for O(-) O(-) (0, 0) (1, 0) (1, 0) (1, 0) (1, 0) Input T (0, 0) (1, 0, 0) T (0, 0) (1, 0, 0)	Recording	g1.rxrec 🗙	3	al heynol - D y Traymage in fancene fandie
BigBalander auf einen BigBalander auf eine BigBalander auf eine BigBalander auf BigBalander auf eine BigB	೫ ៧ ◘ ≡ ७ ୯ ≣	Buckler Maketalan	RE RE	ECORD	RUN RUN	
	Item	Statisticus Statis	🔂 Add n	ew action 🝷	X 🛛 🗋	19 C 🔺
Ramma matida inasaar 🛛 🕯 Si	MobileTest - 🔁 st su	i-D Rogenica 8-09 Recentry Lone.	#	Action		
Servin-Fill JP	🖻 🗹 💽 TestCase					4
C Investory	🛄 🔝 Recording1	Nemen modula Innova 8 2 Security: Fill (2) (i) By Modulater (i) Security (i) Modulater				
		C Investing				
	4					
Reedy						
			4			Compton Units



2

3

Test suite view

This is where you build and control your tests.

The test suite contains a test case with the empty recording module **Recording1.**

🚺 Note

The recording module has a different symbol than usual. This indicates that it's a **mobile recording module**, suitable only for building mobile tests. Standard recording functionality is disabled for them.

All recording modules you create in this project are mobile recording modules.

Recording module view

In the recording module view of **Recording1**, you can start creating your mobile test by adding actions.

4 **Empty actions table** This is where your actions appear.

Create a mobile test

Now that we have our solution, we can start creating the actual mobile test. The procedure is different from that for creating desktop or web tests, but it's not complicated.



Click RECORD.

The dialog for selecting the mobile device and instrumented app appears.

X MobileTesting - Ranorex Studio		- 0 X
File Edik View Project Build Debug Search Tools Window Help		
💽 🗢 📑 🔛 🖓 🗍 400 📷 📬 🔚 🖫 🕼 🛛 vew 🖿 🕅 🖏 🥔 🤨	RUN 🕨 📰 🦓 BULD 🏥 🚠 SPY 🕺 🗱 Defaultisyaat 🗸 🔎	
RECORD	Recording1	× •× c+
Solution MobileTesting	Create a mobile test (2)	م ا
B-monocole resting # Action	Choose a device:	- 1 item
Report Report Assembly/nb.cs		
MobileTestingAting MobileTestingAting MobileTestingAting Add power actions the clicklese	USB-00c18ccf RX iPad	π
Program.cs Program.cs Program.cs button or by draggin button or by draggin		
		0634.06 1 1
	Choose or add app:	
	Web Dropbox	
	🗹 Restart app 🚯 💿 Add app	Ī
	Choose website:	
	www.ranorex.com	1
	Back Create]

Select your device and app. If you select the Ranorex Web Browser, also **specify** the URL for the mobile web test.

Recording1		×	
Create a mobile test	Recording1		×
create a mobile test	Create a mobile test	Recording1	Recording1
Choose a device:	Choose a device:	Create a mobile test	Create a mobile test
i 👗		Choose a device:	
USB-00c18ccf	USB-00c18ccf	USB-00c18ccf RX iPad	Choose a device:
Choose or add app:			USB-00c18ccf RX iPad
	Choose or add app:	Choose or add ann	
Web			Choose or add app:
Restart app	Web Dropbox	Web MiniKeePass	
+	Mestart app	Restart app	Web MiniKeePass
Choose website:	Choose website:	↓ ↓	Restart app 🚯
Park	www.ranorex.com	Choose website:	
Deck	Back	www.ranorex.com	Choose website:
		Back	www.ranores.com
			Deck
a	D	C	0

Depending on your choice, different test types result. In our example:



Mobile Android web tests using the preinstrumented Ranorex Web Browser



Mobile Android app test using the instrumented Dropbox app

Mobile iOS web tests using the preinstrumented Ranorex Web Browser

Mobile iOS app test using the instrumented MiniKeePass app

በ Note

If your app doesn't appear for selection, make sure that you've instrumented it correctly.

Setup mobile test building



Ranorex Studio will automatically carry out the following processes to set up mobile test building:

In Ranorex Studio

Ranorex Studio starts the selected app on your mobile device and adds the respective action in the action table.

🗶 MobileTesting - Ranorex Studio										o ×
File Edit View Project Build	Debug Sea	rch Tools Window	Help							
이 이 🗈 🖬 💾 🖉	ADD 📷	📽 🔓 🖬 🕼	VIEW 💼 🙍 🙃 🛷	📵 RUN 🍖 📰 🍈 BUR	lo 👬 🚠 sev 🗶	Default layou	م · ،			
Projects	# X	Recording1.nevet* X	_							
# EC 12 2 2		RECORD	📔 🚬 RUN	1	🖜 VARIABLES.	. 🖹 SETTINGS	🥻 Studio			
E-Solution MobileTesting		Add new action *	X D D E IV VIA	👻 🥼 Turbo mode 👔 Screensho	e			Search endpoints		
Book Mobile Testing Book References	Acti	on						Desktop	200220	
-B 2 Fg -B Assembly In	Run	mobile app	Current Endpoint	t (USB-00c18ccf5e6b37	03) com.drop	box.android	True		nt nt Windows 19 Pro	
- MobileTesting.rxtst			1						USB-00c18ccf5e6b	3703
H - E Mobile TestingReposito	ryzsrep					No action s	elected	E 🛉	00c18ccf5e6b3703 Connected	
								251460 Comme	30649409e022113a9acabe08 ted	80. 1

The RUN button changes to reflect that it now runs the test on the current automation root, i.e. the active endpoint/mobile devices, not the computer. An action that starts the app on the mobile device has been added to the action table.

On the mobile device

1

2

The instrumented device starts on the mobile device. In our case, this is the Dropbox mobile app.



Attention

Regarding mobile iOS applications, **it is necessary that the Ranorex Service Application is running and active** on the mobile device. Ensure that it is not running in the background as this may prevent you from using the start/stop functionality for the mobile application.

Ranorex Spy

Ranorex Spy starts in live tracking mode for all UI elements of the instrumented mobile app.





Build the test

Building a mobile test consists of two consecutive, iterated steps:



Step 1: Tap, Track, and Add

In this step, you add a UI element as a repository item to Ranorex Studio to make it available for use in actions.



3

Tap: Make the element visible in the app, e.g. by tapping it.

Track: Update the live tracking in Spy by clicking Browse endpoint or Refresh.

Add: Select the element in the element tree, **right-click** it, and **add** it to the repository. Alternatively, **drag and drop** it in the repository.

💃 Ranotac Spy - (12bit) - Live — 🗆 🗙	
p BEMENT	
Content of a data data data data data data data d	Cross Hefe
Tracking OPE Press Cell-UWin for instant tracking	

Step 2: Drag, Drop, and Define

In the second step, you "process" the UI element, i.e. you use it in an action.



2

Drag: From the repository, **drag** the repository item to the desired position in the action table.

Drop: Drop it by releasing the left mouse button. The context menu opens. **Select** the desired action.



Define: Make any necessary configurations to the action, e.g. setting timeouts, content of key sequences, etc. The finished action is shown in the action table, linked to the repository item representing the respective UI element in the app.



በ Note

You can shorten the procedure by directly dragging a UI element from Spy to the action table.



Reference

You can find step-by-step instructions for creating mobile tests in

Web and mobile testing -> Mobile testing > ---> Android app testing example / ---> Android mobile web test example

Web and mobile testing -> Mobile testing > \rightarrow iOS app testing example / \rightarrow iOS mobile web test example

Tips for efficient test building

Remote control your app from Spy

You can remote control your app through Spy. This way, you won't have to carry out all the actions to bring up menus and different screens in the app on the actual mobile device.

12

In the live view of Spy, **right-click** a UI element and **click Touch/Tap.**

The action is executed on your mobile device as if you actually touched the screen.



Enter text in Spy instead

You can also enter text in a similar way.

- 1 2 3
- In Spy, with a UI element selected, **click** the **Advanced** tab.
- Find the Text attribute, enter your text, and confirm with Enter.
 - The text appears in the mobile app as if you typed it in using the screen keyboard.



Run a mobile test

For the most part, running mobile tests works the same as for any other test. There are just a few preparations required.

Preparation

To execute a test on a mobile device, you need to make the following preparations:

- The device must be set as automation root in the endpoints pad. Alternatively, you can specify an endpoint in the **Run mobile app** action. It will then override the automation root.
- The Ranorex Service App must be installed on the mobile device.

Run the test

With the above preparations made, you can run the test like any other.



In the test suite view, **click RUN**. The test will be started and run on the mobile device.

🗶 AndroidMobileAppTest - Ranorex Studio					-		×
File Edit View Project Build Debug	Search Tools Window Help						
💽 O 🖹 🗱 😿 💾 🖉 🛛 ADD	🛤 📽 📴 🔛 🐘 VIEW 🖿 🔟 🙃 🔗 🄇	🕽 RUN 🍺 📰 🏦 🛛 BULO 🏭 .	🛓 SPV 🕺 🏛	Default layout ~	Q		=
Projects # 🗙	Android Mobile App Text and the			y Endp			۹×
부 💌 C (월 📩 🎽		Maintenance mode	MANAGE DATA SOURCES	🌿 Studio	Endpoints	с	+
Solution AndroidMobileApplest B- Solution Items	x / D # 19	Search	h (F3)	ρ Se			٩
AndroidMobileAppTest	Item	Data binding / iterations	Description	Des	ktop		item
References Reports	AndroidMobileAppTest - Test suite				RXLAPTOP020		_
E- Precording modules	B S Test_account				Microsoft Windows 10 Pro		-
 Image: A close Mobile App.nrec 	e [Setup]			And	Iroid		item
B-BA StartMobileApp.nrec	StartMobileApp						
AndmidMobileAnnTest rutmo	- 🔝 TestAccount			Android —			
AndroidMobileAppTest.ntst	i⊟ [TEARDOWN]				LICD 00-10 cef5 of b	2703	, I
- AndroidMobileAppTestReposite	CloseMobileApp				036-00010001000	5/0:	°
- []: app.config - []: AssemblyInfo.cs - []: Program.cs				ŧ= 📳	00c18ccf5e6b3703 Connected		

🛕 Attention

Do not touch the screen or any buttons on the mobile device while the test is running. This would result in a test failure.

በ Note

The various available options for test runs are also the same as for other tests. For more information, refer to \rightarrow Execute a test suite.

Report

Once the test run has finished, the report appears.



The report for mobile tests does not differ from that for other tests.



Android app testing example

In this chapter, we'll go through an Android app test step by step.

Test scenario

We'll test the freely available Dropbox mobile app. Our test contains the following steps:

- 1. Start the Dropbox app on the Android device.
- 2. Check if the Dropbox is empty.
- 3. Create a new folder called MyDocuments.
- 4. Create a screenshot of the Dropbox desktop with the folder in it.
- 5. Delete the folder.
- 6. Check if the Dropbox is empty.

7. Close the app.

Preparations

Device settings

• Apply the required → device settings to your Android device and install the Ranorex Service App.

Connect mobile device

• → Connect your mobile device to your computer, preferably via USB, and → add it as an endpoint in Ranorex Studio.

Instrument the app

Create a Dropbox account

• You'll need a free Dropbox account. Create one at www.dropbox.com.

Create a solution for mobile testing



---> Create a new solution for mobile testing using the RocketStart wizard.

Open the Endpoints pad.

Open the recording module **Recording1.rxrec**.

Ranorex Studio should now look as follows:



- 1 Empty action table in the recording module
- 2
- Empty repository
- 3 Your device in the Endpoints pad.
- 4 Ranorex Service App running on the mobile device, showing the Dropbox app as instrumented.

Create the test



Click RECORD.

The dialog for selecting the mobile device and instrumented app appears.





Select your Android device and app (the Dropbox app) and click Create.

Recording1	×
Create a mobile test	
Choose a device:	
USB-00c18ccf RX iF ad	
Choose or add app:	
Web Dropbox	
Restart app 1 Add ap	p
Choose website:	
www.ranorex.com	~
Back 3 Creat	8

Setup

Ranorex Studio now executes the following processes to set up mobile test building.

In Ranorex Studio

Ranorex Studio starts the selected app on your mobile device and adds the respective action in the action table.

🛠 MobileTesting - Ranorex Studio					-	
File Edit View Project Build D	Debug Search Tools Window	Help				
oy o i 🗈 🖬 💾 🛃	ADD 📷 📬 🔓 🎫 🕼	VIEW 🖿 🔀 🗄 🖗 🤨 RUN 隆 🔲 3	🏟 BURLD 🏭 🚣 SPY 🗶 🛱	🛛 Default layout 🔹 🔎		
Projects	# X Recording1.revec* X					
# RC112 (\$ 14	B RECORD	🔚 📐 RUN 🔢	👟 VARIABLES	🖹 SETTINGS 🌠 Studio		
B-IX Solution MobileTesting	Add new action *	X D C I I V C A V A Turbo mode I	Screenshot			
Mobile Testing	Action Run mobile app	Current Endpoint (USB-00c18ccf5	ie6b3703) com.dropbo	ox.android True	Desktop RXLAPTOP020 Localeott Macrosoft Windows 10 Pro	1 item
	y.csrep	ţ		No action selected	USB-00c18ccf5e60 USB-00c18ccf5e6b3703 Connected	3703 Rohini M



The RUN button changes to reflect that it now runs the test on the current automation root, i.e. the active endpoint/mobile devices, not the computer. An action that starts the app on the mobile device has been added to the action table.

On the mobile device

The instrumented app starts on the mobile device. In our case, this is the Dropbox mobile app.



Ranorex Spy

Ranorex Spy starts in live tracking mode for all UI elements of the instrumented mobile app.





Element browser in Spy showing all the UI elements of the instrumented app.

Live view of the instrumented app on the mobile device.

Build the recording module

Now we can start filling our recording module with actions.

We'll do this using the two-step test-building procedure for mobile tests as explained in detail in ---> Create a mobile test.

Validate that the Dropbox is empty

Step 1: Identify validation text



Click Refresh in Spy.

Drag the UI element for **Empty folder** from Spy to the repository in Ranorex Studio.



Step 2: Define validation



Drag the new repository item to last position in the action table.

In the context menu that opens, click Validation.



Define the Validation action: For Match-Name, select Text and for Match-Value, enter "Empty folder".



Open the menu

2

Step 1: Identify UI element

Ensure that the button for opening the menu is visible on the mobile device.

Click Refresh in Spy.

Drag the UI element for the button from Spy to the repository in Ranorex Studio.



Step 2: Define action

Drag the new repository item to last position in the action table.

In the context menu that opens, click Validation.





በ Note

2

Creating these first two actions illustrates the 2-step procedure, which remains the same for the following actions. Therefore, the following instructions don't describe the procedure in quite as much detail.

Create a new folder

The option to create a new folder is located in the breadcrumb menu of the Dropbox app.



Add the UI element for the menu option New folder to your repository with Spy.

Create a Touch action with this repository item.



Touch action linked to the repository item for the menu option **New folder**.

Enter folder name and create folder

1

2

Add the UI elements for the Folder name text field and the Create button to your repository with Spy.

Create the following three actions with these repository items:

	rch Tools Window Help			
	- 🖼 🕞 📰 📴 🛛 view 💼 🕅 1	🗄 🖉 🤨 RUN 🍡 📰 🏘 BULD 🏭	🚠 SPY 🕺 🏛 Default lay	••
♥₽■★000¢±≤≤0₽ ♥20957	Recording1.nzec X AndroidAppTest.ntst	Recording120180905_122708.mlog		
♥ 💷 ★ ♣ 🛛 🗄	🛑 RECORD 🛛 📔 🐚 RUN		- VARIAB	LE
Dropbox	Add new action * 🗶 🗇 🗖 🔒 🧐	🖓 🔺 👻 அ Turbo mode 🐻 Screenshot		1
	# Action		Comment	
Create new folder	1 Run mobile app Current Endpoi 2 Validate AttributeEqual	Test Empty felder 📑 EmptyFelder		
	@ r	Touch	Deletion 🖃	
MyDocuments	Di Touch	Touch	Kelative 📃 /	Android WidgetEdit lext
2	🕈 6 Set value	Text MyDocum		AndroidWidgetEditText
Canoel Create 3	🗳 7 Touch	Touch	Relative 🔒	Button1
	4 10 Touch Teuch	Relative Hol Delete	<u>†</u>	
Empty folder	I 2 Validate AttributeEqual	Test Empty-folder Empty-Folder		
3 Ustan ents Us dog ments Using ents -4.	13 Onse application CloseWindow	0ms 🔲 ComDrapbosA	ndroid	
and the second second second second	<			>
a qwertyulop	🔕 Add new item = 🗞 Tuck_ 🗶 🗇 📋	🗉 🛛 😌 🗔 🕞 AndroidAppTextRepository = 📼	- Variables 🎸 Cleanup 🔤	1-
, asdfgh)kl 🛃	ten	Path Rese institute/fittle_inst deather to deald?		
`	Control Control Control	Formi Stilles Temphonisticalise Sub-Tempton and Sub-	de l'endant l'icontained II containerture e T	
	📄 📄 AndroidWid	getEditText form[@ti	tle='Create new fol	der']/container[@rid='cor
7123 , 🔘	ab Button1	form[@ti	tle='Create new fol	der']/container[@rid='cor
	Screenshot1			
	AndroidWidgetCheckBox	form(@title="DropboxActionSarActivity")/container(@ri- 1/3/cantainer(@containertype="Linear")/button(@test=	d='content]//container@containertype=1 'Delete']	27
	Button11	form(@title='WyDocuments')/container)@rid='content]/button[@vid='button']]	
	1			



- Touch action to place the cursor in the text field.
- Set value action to enter the folder name.
- Touch action tapping **Create** to create the folder.

O Hint

If you want to test the functionality of the on-screen keyboard, you can also automate entering the name by a series of touch actions on the keys.

Validate folder

Now we'll validate that the folder exists. We'll use image-based validation for this purpose.



Add the folder image to your repository with Spy.

Create the following action with this repository item.





Validation action that uses image-based validation.



Image-based validation is explained in Ranorex Studio fundamentals > Validation > ----> Image-based validation

Delete folder

Deleting the folder requires that we open a menu, tap the first **Delete** button, and then tap the second **Delete** button.



Add the three required UI elements to your repository with Spy.

Create the following actions with this repository item.



- Opens the folder menu by tapping the respective symbol.
- 2 Taps the first **Delete** button.
 - Taps the second **Delete** button.

Validate that the folder has been deleted and close app



3

The repository items that we need already exist. Use them to create the following two actions.
		ch Tools Window Hels	P					
		🛤 🖪 💷 🗛 🔤 🗤	ew 💼 🕅 📾	2 0 L		🕷 BUILD 👯 .A. SPV 🖋	Default lavou	
				•				
		Recording1.orec × And	roidAppTest.ntst R	ecording12018090	5_122708.edog		_	
😌 🖻 🖈 🐥 👘 🗄		RECORD	NUN RUN				🖜 VARIABLE	
Dropbox		Add new action *)C	000000	$ \perp \vee \Rightarrow 1$	urbo mode 👔	Screenshot		
		# Action				0	mment	
		🛛 1 🛛 Run mobile app	Current Endpoint -	com.dropb	True			
		2 Volidate	AttributeEqual	Text	Empty folder	EmptyFolder		
		S Touch	Touch		Relative	MoreOptions		
		🔓 4 Touch	Teuch		Relative	NevFolder		
		§ 5 Touch	Touch		Relative	AndroidWidgetEditText		
		→ 6 Set value	Text	MyDocum		Android///dgetEditText		
		7 Touch	Touch		Relative	Button1		
		EI 8 Validate	Comparelmage	Screenshot1		Android/liidgetImageView		
Emote folder	1 🛛	12 Valida	te /	Attribute	Equal	Empty folder	📄 Emp	tyFolder
Lings y rouser	2 💌	13 Close	applica	CloseWi	ndow		🔲 Com	DropboxAndroid
		 Add new bern • State Add new bern • State ComDropboxAndroi Emstyfolder NoreQuises NoreQuiser 	⊨ X 0 🗂 I	Path Baser /mobileop form(@title="Orn form(@title="Orn form(@title="(no	E AndroidApp (Otitle='con.da opboxActionBarA opboxActionBarA net') and raideler	s®etfRepository = ← Variables & optionandisist] citizty] vontainer[Brid=' contact] (cont of why] to 6 bis minder indetement[Brom ment Jandroi de lement Jandroi de lement]	Cleanop	
		ComDro	opboxAnd	Iroid		Base: /mobileap	o[@title='con	a.dropbox.android'l
4 0 0			A. Calaba			(
	I	Emp	tyrolder			form[@title="Dro	oppoxAction	arActivity j/container[@i
		Button11		form)@title="My	containertype="D Documents")/con	near (/outlon/@text= Delete) tainer)@rid='content']//button(@rid='	button"]	

Validates that the Dropbox home screen is empty (a simple copy of action #2).

Closes the app.

2

Completed recording module

If you followed our instructions, your completed recording should look like this:

🕺 AndroidAppText - Ranorex Studio								n x
File Edit View Project Build Debug Sea	irch Tools Wi	ndow Help						
0 0 🖹 🖬 🔛 🖉 400 🖿	🖬 🗄 🖬		🙂 RUN 隆 📰 🏘 BULD		📰 🛛 Default layout 👻 🔎			1
Projects # X	Recording1.ore	x X AndroidAppTest.ntst Record						ŧх
후 R. C. 입 26 26	RECOR	D 🔚 ⊾ RUN			🖜 VARIABLES	SETTINGS 🌾 Studio		c +
B-B Solution Items	🔂 Add	new action 👻 🔀 🛛	9 C 🗎 🗍 🤊 C		urbo mode	Screenshot		
References Reports AndroidAppText.stmg	#	Action					Comment	
AndroidAppText.otxt AndroidAppTextRepository.rarep	1	Run mobile app	Current Endpoint	com.dropb	True			
- [3: app.config - [3: Assembly/mo.cs	2 🗹	Validate	AttributeEqual	Text	Empty folder	EmptyFolder		
III Program.cs	S 🗳	Touch	Touch		Relative	MoreOptions		
	& 4	Touch	Touch		Relative	NewFolder		
	S 5	Touch	Touch		Relative	AndroidWidgetEditText		
	₽ 6	Set value	Text	MyDocum		AndroidWidgetEditText		
Ranores module browser I X	§ 7	Touch	Touch		Relative	Button1		
Search (F3)	8 🗹	Validate	Comparelmage	Screenshot1		🙀 AndroidWidgetImageView		
Groups	S 9	Touch	Touch		Relative	AndroidWidgetCheckBox		
E Becording1	S 10	Touch	Touch		Relative	ab Delete		
	Sa 11	Touch	Touch		Relative	Button11		
	12	Validate	AttributeEqual	Text	Empty folder	EmptyFolder		
	× 13	Close application	CloseWindow	0ms		ComDropboxAndroid		

You can now switch to the test suite view and run the test. However, in a real-life scenario, the test wouldn't be quite finished here. The next step would be to modularize the recording and structure the test suite.

The concepts of modularizing and structuring apply to all recording modules and test suites, whether for desktop, web, or mobile tests.



Managing recording modules is explained in:

Ranorex Studio fundamentals > Ranorex Studio Overview > ---> Manage recording modules

Structuring a test suite is explained in: Ranorex Studio fundamentals > Ranorex Studio Overview > ---> Test suite structure

A step-by-step example of building a full test suite is available in: Ranorex Studio fundamentals > Ranorex Studio Overview > ---> Build a test

Android mobile web test example

In this chapter, we'll cover the initial steps of a mobile web test on an Android device. After these initial steps, test creation follows the same principles as in the ---> Android app test example and ---> general web test creation.

Preparations

To create a mobile web test, you need to make the following preparations:

Device settings

 Apply the required ---> device settings to your Android device and install the Ranorex Service App.

Connect mobile device

 → Connect your mobile device to your computer, preferably via USB, and → add it as an endpoint in Ranorex Studio.

Install the Ranorex Web Browser app for Android

 Ranorex provides a --> preinstrumented web browser that is required for creating and executing Android web tests.

Choose a mobile browser

Ranorex provides a preinstrumented web browser. You need to use this browser to automate Android web tests.



Open the Endpoints pad.

Open the recording module Recording1.rxrec.

Ranorex Studio should now look as follows:





Empty action table in the recording module

Empty repository

Your device in the endpoints pad

Ranorex Service App running on the mobile device, showing the preinstrumented 4 web browser app. If you chose a different web browser and instrumented it correctly, it will also appear here.

Create the test



Click RECORD.

The dialog for selecting the mobile device and instrumented app appears.





Select your Android device app (web browser), specify the URL, and click Create.

Setup

Ranorex Studio now executes the following processes to set up mobile test building:

In Ranorex Studio

Ranorex Studio starts the selected app on your mobile device and adds the respective actions in the action table.



- The RUN button changes to reflect that it now runs the test on the current automation root, i.e. the active endpoint/mobile devices, not the computer.
- 2 An action that starts the app on the mobile device has been added to the action table.
- 3 An action that navigates to the specified URL in the selected browser app has also been added.

On the mobile device

Ranorex Studio starts the selected browser and navigates to the specified URL on the mobile device.



Ranorex Spy

Ranorex Spy starts in live tracking mode for all UI elements of the instrumented mobile app.





Element browser in Spy showing all the UI elements of the instrumented app.



Live view of the instrumented app on the mobile device.

Build the test

From this point forward, test building follows the same principles as in the ---> Android app test example and ---> general web test creation.

iOS app testing example

In this chapter, we'll go through an iOS app test step by step.

Test scenario

We'll test the preinstrumented iOS KeyPass mobile app that you can download from the Ranorex website. Our test contains the following steps:

- 1. Start the app on the iOS device.
- 2. Create a new password database.
- 3. Name it MyPasswords.
- 4. Give it the password 1234.
- 5. Check if the category General exists.
- 6. Exit the app.

Preparations

Device settings

• Apply the required ----> device settings to your iOS device and install the Ranorex Service App.

Connect mobile device

• → Connect your mobile device to your computer, preferably via USB, and → add it as an endpoint in Ranorex Studio.

Instrument the app

• Download the ---> preinstrumented KeyPass app from our mobile download archive and install it on your device.

Create a solution for mobile testing



---> Create a new solution for mobile testing using the RocketStart wizard.

Open the Endpoints pad.

Open the recording module Recording1.rxrec.

Ranorex Studio should now look as follows:



- Empty action table in the recording module
- 2 Empty repository
- 3 Your device in the Endpoints pad.
 - Ranorex Service App running on the mobile device, showing the KeyPass app as instrumented.

Create the test



1

4

Click RECORD.

The dialog for selecting the mobile device and instrumented app appears.

🛠 MabileTesting - Ranorez Studio		- 🗆 X
File Edit View Project Build Debug Search Tools Window Help		
💽 🕤 📴 🌃 🔛 🖉 ADD 📷 😭 📴 📴 🚺 VEW 🖿 💆 🕄 🖉 🤤	RUN 🕨 🔳 🎆 BUILD 🏭 🚠 SPY 🎇 🏛 🛛 Default layout 👻 🔎	
RECORD	Recording1	× **
B ■ Solution MobileTesting B = Solution Items	Create a mobile test	2 🖉
Mobile Testing # Action	Choose a device:	- 1 item
	USB-OOc18ccf RX iPad	म 1 Ren म
ON Recording Lovec button or by draggin		1 item
		063fc8+ F
	Choose or add app:	
	Web Dropbox	
	Restart app 🚯	Add app
	Choose website:	
	www.ranorex.com	~
	Back	Create
Select your iOS device and a	app (KeyPass app) and click Create.	
Recording1	×	
Create a mobile test		

Create a mobile tes	at	
Choose a device:		
USB-00c18ccf	RX iPad	
Choose or add app:		
Web	MiniKeePass	
🗹 Restart app 🚯		Add app
Choose website:		
www.ranorex.com		~
Back		3 → Create

Setup

Ranorex Studio now executes the following processes to set up mobile test building.

In Ranorex Studio

Ranorex Studio starts the selected app on your mobile device and adds the respective action in the action table.

🛠 iOSAppTesting - Ranorex Studio	- 0	×
File Edit View Project Build Debug Search Tools Window Help		
💿 🖓 💿 📑 📸 💾 🥙 add 📷 📬 📴 🚱 🕼 🖓 Iew 🖿 💆 🗟 🔗 🤨 Run 🍖 🔲 🕷 build 🗄	🗄 🚠 SPY 🔏 🞞 🛛 Default layout 🗸	ٍ م
Projects # X Recording Lourect X	₽ Endpoints	# ×
🛎 🖻 C 🖺 🚴 🥔 👩 RECORD 🔚 🖪 KUN RIALES 🖹 SETTINGS 🏌 Stud	o Endpoints	c +
□ R Solution iOSAppTesting ⊕ Solution items △ Add new action ▼ ※ □ □ ■ ※ ○ ▲ ▼ => Turbo mode ■ Screenshot	Search endpoints	P
B # Action	Desktop	1 item
2 - Action 1 Run mobile app Current Endpoint (RX iPad) ranorex.MiniKeePass True	RXLAPTOP020 Localhost Microsoft Windows 10 Pro	72
	ios	1 item
- Mai IOSAppTesting.rxtst	1	item
	(iPad 14dd96649409e022113a9acabe0 prinected	7

The RUN button changes to reflect that it now runs the test on the current automation root, i.e. the active endpoint/mobile devices, not the computer. An action that starts the app on the mobile device has been added to the action table.

On the mobile device

1

2

The instrumented app starts on the mobile device. In our case, this is the KeyPass mobile app.

iPad 👻	11:42	\$ 96 % 💻 +
	Files	
	Tan the + button to add a new KeeDass file	
	top the + button to out a new reepassine.	
ക	۲	+

Ranorex Spy

Ranorex Spy starts in live tracking mode for all UI elements of the instrumented mobile app.

🛠 Ranorex Spy - (32bit) - Live	- o ×
	🖹 SETTINGS 🥢 Studi
TRACK [mobileapp[@title='ranorex.MiniKeePass']	
BROWSER & RESULTS PATH EDITOR	
	pp 'ranorex.MiniKeePass' ‡∐ 768 x 1024
MobileDevice 'RX iPad' MobileApp 'ranorex.MiniKeePass'	'ranorex.MiniKeePass' 2
 □··· □ Form 'UIWindow' □··· □ Container 'LayoutContainer' □··· □ Container 'TextEffectsWindow' 	Nes.
Мор	



Element browser in Spy showing all the UI elements of the instrumented 100app.

Live view of the instrumented app on the mobile device.

Build the recording module

Now we can start filling our recording module with actions.

We'll do this using the two-step test-building procedure for mobile tests as explained in detail in ---> Create a mobile test.

Create new password database

Step 1: Identify UI element and add to repository



Ensure that the + button is visible on the mobile device.

Click Refresh in Spy.

Drag the UI element for **Empty folder** from Spy to the repository in Ranorex Studio.

			🗙 khitepylheding - Ransons Stratio — 🗆 💥
			File Edit New Project Suit Debug Search Tools Mindon Help
			💽 O 📑 副 🖬 🔐 🖉 A 2 🍇 🤓 🎉 🗔 🍇 💷 🖄 🔤 🏛 🖄 🖉 🗑 A 4 🗽 🖷 🗮 A 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
			Pagenta # X Reconfigurent X Orlag Technolog - T Endpoints # 3
			2 P. C 12 2 2 4 Ferner C +
			Control Object integr D Address adds - X D B 9 9 A V A Subscrade Exceeded
	ria: Film	1 (1) ((1) (((1) (((1) (((1) ((((() (((() (((() ((((Allow Polytop and Arline Develop 1 deve
		P Description (Chick Street	Room Chine Report (K: Pau) Index Tage Current Endpoint (K: Pau) Index MinikaPau Tau
-		Ranorex Spy - (S2Dit) - Live	The second
		ELEMENT	Collegitationships
		> BROWSUK	+ EX Kyg/tering/gapatogram
		TRACK /mobileapp@title='rance	arec Min as distancing tare
			Add now item - @ Task 97 E P = 10 O E Search (63)
		BROWSER & RESULTS PATH EDITOR	
			Path
	The first holder is self a new feedbard for		Banaraw Minika Para
		Device of Fad	Base: /mobileapp[@title= fanorex.minikeePass]
		B- D MobileApp 'ranores.Min KeePass'	Term////toolbar[@accessibilityid='Toolbar]//cont
		Container 'LavoutContainer'	S Recording1 🛒
		B-Container WewControllerWy	apper'
		B TitleBar Files'	LocalizationKay
		in Container / UlBarBackgr	reverd 2 2 weet
	-	Gontainer UIToolbarCo	InteritView Consultation
	•	Container Ulliuttor	BarStackView constraints accessibility hint specified for the iOS control.
		B- + trainer help'	
			pdernBarButton'
	0		ainer Add
	<u> </u>	ab B	utton' UIModernBarButton'
		i⊞- 🛄 Con	

Step 2: Define action

4

5

6

- **Drag** the new repository item to last position in the action table.
- In the context menu that opens, **click Touch**.
- The Touch action needs no further definition and is complete.



በ Note

Creating this action illustrates the 2-step procedure. It remains the same for the following actions. Therefore, the following instructions don't describe the procedure in quite as much detail.

Enter the database name

After tapping the + button, a dialog opens asking for a name and a password for the new database. To enter the name, we'll need to create two actions.

12

Add the name text field to your repository with Spy.

Create the following two actions with this repository item:





Touch action that places the cursor in the text field.

Set value action that enters the name.

O Hint

If you want to test the functionality of the on-screen keyboard, you can also implement entering the name by a series of touch actions on the keys.

Enter and confirm the database password



Add the two text fields for the password and its confirmation to your repository with Spy.

Create the following four actions with these repository items:



- Touch action and Set value action to place the cursor in the text field and enter the password.
- 2 Touch action and Set value action to place the cursor in the text field and repeat the password.

Confirm database creation

2

Add the Done button to your repository with Spy.

Create the following action with this repository item:



Touch action that taps the Done button to confirm database creation.

Validate database file

Now we'll validate whether the database was created correctly.



Add the file name to your repository with Spy.

Create the following action with this repository item:



Validates that the displayed file name is the same as the one entered during database creation.

Delete database and close app



Add the Edit, (-) and Delete buttons to to your repository with Spy.

Create the following actions with these repository items:

			iOSApp	Testing.ntst Recor					÷
				RECORD 🛛 🔚	NUN RUN		👟 VARIABLES	🖹 SETTINGS	🌾 Studio
			🔾 Add	new action 🔹 🛛 🛞	≜ 9 ¢ ≜	🐨 🛛 At Turbo mode	Screenshot		
**	a att a 10.5 and 1			Action					c
	* 100 %		🖸 1	Run mobile app	Current Endpoint (RX iP	ranorex.MiniKee	True		
	· · · · · · · · · · · · · · · · · · ·		Sec. 2	Touch	Touch		Relative	🐱 ButtonUIMod	ernBarButt
			S 🔒	Touch	Touch		Relative	Name	
	Edit		Ψ4	Set value	Text	MyPasswords		Name	
			S 5	Touch	Touch		Relative	Password	
			P 6	Set value	Text	1234		Password	
	Databases		Sec. 7	Touch	Touch		Relative	ConfirmPassw	ord
			P 8	Set value	Text	1234		ConfirmPassw	ord
	MyPasswords.kdb	_	<u>8</u> 9	Touch	Touch		Relative	Done 🗠	
	Last Modified: 07.09.18, 05:17	1	Se 11	Touch	Touch	Rela	tive	💩 Edit	
	Dilut	2	Se 12	Touch	Touch	Rela	tive	ab Delete	e
	Delete	3	🆺 <u>13</u>	Touch	Touch	Rela	tive	🗖 UlSwi	peAction
		4	× 14	Close app	lic CloseWindo	w		🔲 Ranoi	rexMiniK
. 6		_	O Add	new item + % Trac	k. X 🛛 🗖 🖩 9	6. E8	Search	(F3)	,p. =
	• •		Iten	RanorexMiniKeePas	5	Path Base: /mobileap	p]@title='ranorex.Mir	iKeePass]	
_		r	ab	Edit	?/?/ti	tlebar[@acce	ssibilityid='	Files']/contair	ner[3]//bu
	Sp-		ab	Delete	.//tab	ole[@accessib	oilitylabel='E	mpty list']/?/?	/button[
				UISwipeAc	tionPullVie ,//tab	ole@accessib	oilitvlabel='E	mptv list'1/co	ntaiņer[2

- 1 Touch action that taps **Edit**.
- 2 Touch action that taps (-).
- 3 Touch action that taps **Delete** to confirm deletion.
- 4 Closes the app.

Completed recording module

If you followed our instructions, your completed recording should look like this:

VARIABLES	BULD ## SPV X Default layout ~ Endpoints S & SETTINGS Konstantion	۵ Endpoints	; • x
vARIABLES 19 (≈ 🔺 🐨 斗 Turbo mode	Endpoints	Endpoints	
9 🤉 🔺 🔻 🕸 Turbo mode			c +
Idpoint (RX iP ranorex.MiniKee T MyPasswords 1234 1234 ontains AccessibilityLabel M	Screenshot True Relative Relative Relative Relative ConfirmPassword Relative ConfirmPassword Relative Relative Done MyPasswords Relative Elit Relative Done MyPasswords Relative Done ConfirmPassword Relative Done ConfirmPassword Relative Done Done Done	C PP020 M Windows 10 Pro 6649409eb22113a9ac xd	P 3 item 1 item
o	MyPasswords 1234 1234 Intains AccessibilityLabel	Relative B utton/UlModernBarButt Relative Name MyPasswords Name 1234 Password 1234 ConfirmPassword 1234 ConfirmPassword 1234 ConfirmPassword 1234 ConfirmPassword Relative Done Intains AccessibilityLabel MyPasswords.kdb MyPasswords Relative Delete Relative Delete Relative UlSwipeActionPullView ow Oms RanorexMiniKeePass	Relative B uttonUlModernBarButt Relative MyPasswords Name 6669409c02211399ac Relative Password 1234 Password Relative ConfirmPassword 1234 ConfirmPassword Relative Done Intains AccessibilityLabel MyPasswords.kdb MyPasswords Relative Done Relative Done Relative Done Relative Dolete Relative UlSwipeActionPullView Relative DanporexMiniKeePass

You can now switch to the test suite view and run the test. However, in a real-life scenario, the test wouldn't be quite finished here. The next step would be to modularize the recording and structure the test suite.

The concepts of modularizing and structuring apply to all recording modules and test suites regardless of desktop, web, or mobile tests.



iOS mobile web test example

In this chapter, we'll cover the initial steps of a mobile web test. After these initial steps, test creation follows the same principles as in the \rightarrow iOS app test example and \rightarrow general web test creation.

Preparations

To create a mobile web test, make the following preparations:

Device settings

 Apply the required ---> device settings to your iOS device and install the Ranorex Service App.

Connect mobile device

 → Connect your mobile device to your computer, preferably via USB, and → add it as an endpoint in Ranorex Studio.

Install the Ranorex Web Browser app for iOS

• Ranorex provides a ---> preinstrumented web browser that is required for creating and executing iOS web tests.

Create a solution for mobile testing



---> Create a new solution for mobile testing using the RocketStart wizard.

Open the Endpoints pad.

Open the recording module **Recording1.rxrec**.

Ranorex Studio should now look as follows:



- 1 Empty action table in the recording module
- 2 Empty repository
- 3 Your device in the Endpoints pad
- 4 Ranorex Service App running on the mobile device, showing the preinstrumented web browser app.

Create the test



Click RECORD.

The dialog for selecting the mobile device and instrumented app appears.

🗙 MabileTesting - Ranorex Studio		- D X
File Edik View Project Build Debug Search Tools Window Help		
💽 🖸 📴 📸 🖓 🗛 🗛 🖓 🔓 🖓 🕼	🗓 🕄 🖉 🙂 RUN 🕨 🔳 🆓 RUND 🏭 🚣 SAV 🔏 🗮 🔤	ult layout v 🔎 🔎 .
Projects	Recording1	× **
Solution MobileTesting	Create a mobile test	2 🔎
H-References # Action	Choose a device:	1 tem
- Carlorita - Carlorita - Date app. config - Date app. config		1 Rem
- 199 Mobile lesing acting Mobile TestingRepository.zxnep Add news	actions by dicking USB-00c18ccf RX iPad	
Orgram.cs Determing1.xxec but	tton or by draggin	
		053/cs ag
	Choose or add app:	
	Web Dropbox	
	Restart app	Add app
	Choose website:	
	www.ranorex.com	✓
	Back	Create

Select your iOS device, **select** the app (Ranorex Web Browser), **specify** the URL, and **click Create**.



Setup

3

Ranorex Studio now executes the following processes to set up mobile test building:

In Ranorex Studio

Ranorex Studio starts the selected app on your mobile device and adds the respective actions in the action table.

🜿 iOSWebTest - Ranorey Studio									1 X
Ele Edit View Desired Ruild	Debug	Course Tools Wesdam Hide				_	_		
File Eart view Project build	Debug 3	search tools window heig							
07 0 💽 🖬 💾 🖉		🖿 📽 🖻 🖪 🚺 🖤	en 📴 🗃 😰 🤌	🤨 RUN 隆 🗐 🕷	8 BUILD 🏭 🚣	ISPY 🕺 🗏 🗄	Default l	eyout ~ 🔎	
Projects	• x	Recording1.nrec* ×							# ×
		🛢 RECORD		VARIABLES_	SETTINGS	🕼 Studio		Endpoints	c +
Solution IUSWebTest									Q
🖃 💀 iOSWebTest 🚭	🕽 Add n	ew action 👻 🗶	9 11 11 19	(? ▲▼	🕸 Turbo mode	D Se	Desktop		
References	÷	Action						XLAPTOP020	
-Ca app.config		ACTON						ficrosoft Windows 10 Pro	
Assembly 2	1	Run mobile app	Current Endpo	oint (RX iPad)	ranorex.RxBro	wser	ios		
- iOSWebTr 3 t	2	Set value	PageUrl		www.ranorex.	com		X iPad 514dd96649409e022113a9ae	
B-E iOSWebTesmep	-	Sectore	rugeon				- - 2		
- D Program.cs			t						_
E- dA Recording Lintrec	- L				i05				item
1	- H	C				DV IDad			I
		🕽 Add new item 👻 🔍 Track	X 🗇 🗂 🖬 🧐 🤗	Searc	💴 🗖 🖃		00000000	000110.0	_ II
Ranorex module browser	I X	Item		Path	- 번드 (트)	2514dd	90049409	e022113a9ac	73
Search (F3)	Q	B- D Browser		Base: /mobileapp(@title='ran	iorex.P				
□ → R iOSWebTest		Dom		Base: /mobileapp[@title='ran	iorex.P				

The RUN button changes to reflect that it now runs the test on the current automation root, i.e. the active endpoint/mobile devices, not the computer.

- 2 An action that starts the app on the mobile device has been added to the action table.
- 3 An action that navigates to the specified URL in the web browser app has also been added.

On the mobile device

Ranorex Studio starts the preinstrumented web browser and navigates to the specified URL on the mobile device.



Ranorex Spy

Ranorex Spy starts in live tracking mode for all UI elements of the instrumented mobile app.



Element browser in Spy showing all the UI elements of the instrumented app.

Live view of the instrumented app on the mobile device.

Build the test

2

From here on out, test building follows the same principles as in the \rightarrow iOS app test example and \rightarrow general web test creation.

Advanced mobile testing

iOS Service App

This quick starting guide will show you, how to properly use the new iOS RxServiceApp to create automated tests and how to prepare your iOS project so that start/stop app functionality can be used.

Prepare Your iOS Project to Enable Start/Stop Functionality

The following step by step instructions are based on Xcode 6.

If the app will be instrumented manually, please do the following steps to register the custom URL scheme so that the starting and stopping will work correctly:

- Open your project in Xcode 6.x
- Follow the instructions in the section: Instrumentation with source code on iOS
- In Xcode choose the "Info" tab
- Under Custome iOS Target Properties add URL types key
- Expand URL types key with Item0
- Expand Item0 with URL schemes
- Expand URL schemes with Item0 and write the app activity name as value

Gene	ral Capabilities Info	Build Se	ettings	Build Phases	Build Rules
▼ Custom iOS Tai	rget Properties				
	Key		Туре	Value	
	Bundle versions string, short	÷	String	1.0	
	▼ URL types	ŧ	Array	(1 item)	
	▼ltem 0		Dictionary	(1 item)	
	WURL Schemes	ŧ	Array	(1 item)	
	ltem 0	00	String	💲 ranorex.demo	papplication
	Bundle identifier	ŧ	String	JZ6752U8V8.	<pre>\${PRODUCT_NAME:rfc1034identifier</pre>
	InfoDictionary version	ŧ	String	6.0	
	▶ Icon files (iOS 5)	ŧ	Dictionary	(1 item)	
	Bundle version	\$	String	1.0	
	Main nib file base name (iPad)	\$	String	MainWindow	
	Executable file	\$	String	\${EXECUTABL	E_NAME}
	Application requires iPhone environ	ment 🛊	Boolean	YES	÷

🕕 Note

- If no URL scheme is set, the app under test cannot be launched and you will get an error message when you try to start the app with Ranorex. In the app list of the iOS RxServiceApp there will be also a warning message "Can't launch".
- After Instrumenting and deploying the app for the first time, start the app manually once.
- The newly instrumented and deployed app, needs to be started manually for the first time, so that it gets registered at the RxServiceApp.
- If the app is instrumented with Ranorex Studio on the Windows machine, the correct URL scheme will be set automatically.
- Due to restriction in Apple API it may occur that the instrumented IPA is not able to run from with Ranorex Studio, while RxServices application is in the background mode. It is recommended to perform any test while the instrumented IPA is in the foreground

Proper use of the iOS Service App

To use the new start/stop functionality via USB or WiFi, make sure that the RxServiceApp is running and is active on the device. You can either use the instrumentation wizard to deploy the service app as described in ---> Instrumentation Wizard – iOS, or download and install the service app on your device from our mobile download archive by scanning the following QR Code or the using the following short URL:



Use the QR Code above or the url https://bitly.com/mArchive to directly access the download archive on your mobile device.

1 Note

Start/Stop functionality can not be used from the home screen of the device, even if the RxServiceApp is running in the background.

You can still create automated tests without the iOS RxServiceApp. Please note that without the RxServiceApp the start/stop functionality won't work.

To add a device without the service app, make sure that an instrumented app is running and active on the device.

1 Note

- When using the start functionality, the app under test will be launched multiple times, because it needs to be reseted into initial state.
- If you want to create automated test on an instrumented app that has no URL Scheme, please make sure, that the RxServiceApp is not installed on the device and that the testing app is active on the device.

Non-UI Testing

Next to automate testing of the user interface of mobile applications it's also possible to perform non-UI tests invoking technology dependent actions (see -->Detailed list of actions). This feature allows you to get information about the devices hardware, the operating system etc.

Non-UI testing on Android

You are able to get the device info of your Android device directly by right-clicking the mobile device in Ranorex Spy.



You can access the non-UI testing methods in the action table by adding an invoke action on the mobile app.



You can also access the non-UI testing methods within User Code Actions as well as Code Modules.

The code should look something like the following:



```
// report the manufacturer of the device
```

Report.Info("Calls: " + calls.ToString());

VB.NET

```
Dim app = repo.AndroidApp.Self.[As](Of AndroidApp)()
' get an AndroidDeviceInfo object holding all available device info
Dim info = app.GetDeviceInfo()
' get a list of SMS from the device
Dim sms = app.GetSms()
' get a list of calls from the device
Dim calls = app.GetCalls()
' report the manufacturer of the device
Report.Info("Manufacturer: " & info.Manufacturer)
' report the manufacturer of the device
Report.Info("SMS: " & sms.ToString())
' report the manufacturer of the device
Report.Info("Calls: " & calls.ToString())
```



For further information about code modules and user code actions please have a look at the chapters --->Code modules and --->User code actions.

Cross-device mobile tests

In this chapter, we'll show you how you can configure one test to run on several different devices of the **same mobile OS (Android OR iOS)** consecutively and even in parallel.

The configuration is based on data-driven testing and is **the same for both Android and iOS**. For the sake of simplicity, the screenshots refer to Android only. በ Note

Make sure to add a Close application action when running your test on different devices. Otherwise, the tests may run in a different order or be skipped for some devices.

Recordin	ng1.rxrec* ×									÷
R		NUN					- v/	ARIABLES	🛓 SETTINGS	🎾 Studio
C Add	new action = 1 %	n m≏ino o		- Turbo mode	Screensh	at				
 Add 1 # 1 2 3 4 5 6 7 8 9 10 11 	Action • 🗶 🚺 Action Run mobile app Touch Touch Set value Touch Wait for Validate Get value Report Invoke action Mobile key press	Image: Constraint of the second se	ementlocati	Image: state of the state o	Screenshi keepass T F F	irue kelative kelative kelative Mail	TextStorageEmula TextStorageEmula Password PassOk TextStorageEmula EMail EMail List ComAndroidKeep	ted ted	No action sel	ected
<	new item	ed	Image: Second secon	ch pe gesture bile key press dation oke action value device orientation se application t for	epositor id.keepa [@rid='p @rid='lis ton[@res	y + += Varial C 55'] assword'] t)/text[@ ourceid=	oles 💉 Cleanup omment	> Sear	ch (F3) 🔁 Update scre	eenshot

Configure a consecutive cross-device test

- **Open** the first recording module in your test. It should contain the Run mobile app action.
- Click the Endpoint property and click As new variable...



Name the variable **varMobileDevice**. This variable will control which device the test app will be started on, and therefore where the test will be run.

Recording1.rxrec* ×											
	RI	ECORD	RUN							•	- VARIA
•	\dd n	ew action 👻 🗍	📋 🛢 9 🤉 🔺 🔻 💷 T	urb	o mode	G Screens	hot				
#		Action	Endpoint	Sta	artup argui	ments	Restart ap	p			
	1	Run mobile app	Nexus 5X 🗸	cor	m.android.	keepas 🗸	True	~]		
£ 2	2	Touch	Current Endpoint (Nexus 5X)				Relative		📄 Te	xtStorageE	mulated
£ :	3	Touch	Nexus 5X		Add varia	able					\times
Ψ	4	Set value	As new variable	rx							
£ :	5	Touch	avariext Touch		Variable	name:	varMobile	eDevice			
0	5	Wait for	NotExists	5s	Default	/alue:	Nexus 5X				
	7	Validate	AttributeEqual	Te							
₽8	3	Get value	Text	۶v				OK		Cance	el
b 9	9	Report	Log	\$v							



Right-click the test suite that contains the recording with the Run mobile app action and **click Data source...**





5

In the dialog that opens, **click New > Simple data table**.

TestCase properties	_		×
TestCase			
General Data source Data binding Condition			
New 🔽 No data source 🗸 🗸	Manage data	sources	
Simple data table			
CSV connector			
Excel connector			
SQL connector			
No data source selected			
Click New to add a new data source			
Data range		Refres	h
All rows Range	D : ((incires and incires	
	Preview effec	tive data s	et
ОК	Cancel	Арр	ly

7

Fill the data table with the names of the desired devices as they appear in the endpoints pad.

TestCase properties	_		×
TestCase			
General Data source Data binding Condition			
New 🛛 🕶 Mobile_Devices - SimpleDataConnector 🗸 🗸	Manage d	ata sources	
colMobileDevice Add column 1 Nexus 5X 2 GT-P7500 3 Add row			
Data range		Refre	۶h
e.g. 1-5, 8, 11-13	Preview ef	fective data :	set
ОК	Cancel	Арр	oly .::

8 Click Data binding and under Module variable, bind the data column with the device names to the device variable varMobileDevice.

stCase	properties estCase				-			>
eneral	Data source	Data binding	Condition					
Variab	le binding							
Data	column			Module variable				
colMo	bileDevice			Recording1.varM	lobileDevice			\sim
				Recording1.var	r l ext rMobileDevice			
					Auto-bind	Cle	ar bind	ings
Daram	ators				Auto-bind	CIE		ings fo
	Name		Value		Module var	iable		
▶1	Add row							\sim

9 The test will now be iterated for each device in your data source. This will also be reflected in your report.

Ranorex			
Android			
Computer/Endpoint Nexus 5X (Unknown) USB - 00c18ccf5e6b3703	Execution time 30/11/2017 10:41:47	Test case result summar	y
Operating system - OS Language - Total errors 0	Screen dimensions - Duration 15.58s Total warnings 0	2x Success	
Expand test containers Expand details Test container filter: Success Fa	Collapse all	-	
TestCase Rows: 2			15.55s
TestCase Data Row: 1			8.78s
 Test data colMobileDevice: Nexus 5X 			
▶ O Recording1			8.73s
TestCase Data Row: 2			6.75s
 Test data colMobileDevice: GT-P7500 			

Configure a parallel cross-device test

It's also possible to run one test on multiple devices at the same time.



First:

a

Configure a variable as in steps 1-3 in the instructions for a consecutive cross-device test above.



Or simply **disable** the existing data source if you're working with the test from the instructions for a consecutive cross-device test above.

TestCase properties	_		×
TestCase			
General Data source Data binding Condition			
New No data source Mobile_Devices - SimpleDataConnector No data source No data source selected Click 'New' to add a new data source	Manage data	sources	
Data range All rows Range		Refres	h
	Preview effect	tive data s	et
ОК	Cancel	Арр	ly



In the test suite view, **right-click** the test suite node and **click Global parameters...** Add a global parameter called **globalMobileDevice**.

Android General Global parameters Parameters I globalMobileDevice Nexus 5X (Unbound) > 2 Add row	droid p	oroperties					_		×
General Global parameters Report Parameters I globalMobileDevice Nexus 5X (Unbound) > 2 Add row Image: Comparison of the second secon	D A	ndroid							
Parameters Name Value Module variable 1 globalMobileDevice Nexus 5X (Unbound) > 2 Add row Image: Constraint of the second	eneral	Global parameters	Report						
Name Value Module variable 1 globalMobileDevice Nexus 5X (Unbound) > 2 Add row	Param	eters .						Inf	fo
1 globalMobileDevice Nexus 5X (Unbound) ▶ 2 Add row		Name		Value		Module	variable		
Add row	1	globalMobileDevic	e	Nexus 5X		(Unbour	nd)		\sim
Auto-create Auto-bind Clear b	▶2	Add row							\sim
Auto-create Auto-bind Clear b									
Auto-create Auto-bind Clear b									
Auto-create Auto-bind Clear b									
Auto-create Auto-bind Clear b									
Auto-create Auto-bind Clear b									
Auto-create Auto-bind Clear b									
Auto-create Auto-bind Clear b									
Auto-create Auto-bind Clear b									
Auto-create Auto-bind Clear b									
Auto-create Auto-bind Clear b									
Auto-create Auto-bind Clear b									
Auto-create Auto-bind Clear b									
Auto-create Auto-bind Clear b									
Auto-create Auto-bind Clear b									
Auto-create Auto-bind Clear b									
Auto-create Auto-bind Clear b									
Auto-create Auto-bind Clear b									
					Auto-create	Auto-bin	d Cle	ear bindi	ngs
OK Cancel					OK	Can	cel	App	lv

4 5 **Right-click** the test case that contains the recording where you specified the variable and **click Data binding...**

Under **Parameters > Module variable, bind** the variable to the global parameter.

incuse	properties					>
D Te	estCase					
eneral	Data source Data binding	Condition				
Variab	le binding					
Data	column		Module variable			
				Auto-bind	Clear bi	ndings
Param	neters					Info –
	Name	Value		Module vari	iable	
		AL			A A DESCRIPTION	-
▶1	Android.globalMobileDevice	Nexus 5X		Recording1.	/ariviobile	D ~
▶1 2	Android.globalMobileDevice Add row	Nexus 5X		Recording1.	1.varMobile	D ✓ eDevice
▶1 2	Android.globalMobileDevice Add row	Nexus 5X		Recording1.	1.varMobil	D V eDevice
▶1 2	Android.globalMobileDevice Add row	Nexus 5X		Recording	7ariviobile 1.varMobil	D ∽ eDevice
▶ 1 2	Android.globalMobileDevice Add row	Nexus 5X		Recording	7ariviobile 1.varMobil	D ∽ eDevice
▶ 1 2	Android.globalMobileDevice Add row	Nexus 5X		Recording	/ariviobile 1.varMobil	D ∽ eDevice
▶ 1 2	Android.globalMobileDevice Add row	Nexus 5X		Recording1.	/artviobile 1.varMobil	D ❤ eDevice
▶ 1 2	Android.globalMobileDevice Add row	Nexus 5X		Recording1.	2 arviobile 1.varMobil	D ∨ eDevice
▶ 1 2	Android.globalMobileDevice Add row	Nexus 5X		Recording1.	2 IIVIODIIE 1.varMobik	D ∨ eDevice
▶ 1 2	Android.globalMobileDevice Add row	Nexus 5X		Recording	1.varMobil	D V
▶ 1 2	Android.globalMobileDevice Add row	Nexus 5X		Recording	1.varMobile	D V eDevice
▶ 1 2	Android.globalMobileDevice Add row	Nexus 5X	Auto-create	Recording Recording	Clear bi	D V eDevice
▶ 1 2	Android.globalMobileDevice Add row	Nexus 5X	Auto-create	Recording Recording	Clear bi	ndings

In the repository, **find** the app folder for your mobile app and **click EDIT IN SPY**.

6

🔁 Add new item 👻 🄍 Track 🛠	0011	🖓 🔄 📰 TestSolutionRepository 🕶 🛏 Variables	🍼 Cleanup
Item		Path	Comment
🗷 ··· 📘 ComAndroidKeepass	🗶 EDIT IN SPY	Base: /mobileapp[@title='com.android.keepass']	
	Cli	ck to edit path in Spy	


8

Modify the path as in the screenshot below and ensure the attribute **devicename** has the value **\$varMobileDevice**, i.e. the device variable specified earlier.

🔀 Editing 'ComAndroidKeepass'	– 🗆 X
	🕒 SETTINGS 🌾 Studio
(mobileapp[@title='com.android.keepass' and @	Odevicename=\$varMobileDevice] APPLY
C Refresh 🗔 Highlight	▶ General
M ☐ mobileapp(@title='com.android.keepass' and @devicename=\$varMobile[.	▲ MobileApp ☑ title = ∨ com.android.keepass ∨ ☑ deviceoname = ∨ SvarMobileDevice ☑ deviceorientation = Nexus 5X △ name = As new variable ○ osversion = SvarMobileDevice ○ versioncode = (2 distinct values) ○ versionname = (2 distinct values)
Click 'Refresh' to update result elements	
One element found for current selection.	,

To run your test in parallel, start it from a command line environment with the following arguments:

```
start MobileTest.exe /pa:globalMobileDevice="Galaxy Nexus"
start MobileTest.exe /pa:globalMobileDevice="GT-P7500"
```



Automate Android system apps

When automating an Android App it might sometimes be necessary to leave the application under test. For example, you want to:

• check if an issued notification has been received

- share something on a social network
- change the system settings
- ...

To automate these tasks, you need to automate the Android OS. This comes with certain special requirements and restrictions that are explained in this chapter.

 Note To automate system enabled. The setting settings in>setting To speed up the test generation of screet section of the plugi 	n apps the setting 'And g can be found in the r gs dialog. t execution of system nshots for android. Th n specific settings in	droid OS Automation' has to be mobile section of the plugin spec app automation you can disable be setting can be found in the mo	cific e the obile
Settings		×	
The Contract			
Settings			
General Advanced Recorder defaults Repository of	lefaults Imaging Plugins		
User settings	Solution settings		
Use Java SWT legacy automation mode	False		
Whitelisted class names	Tube		
Y <mark> Mobile</mark>			
Android OS automation	True		
Connect timeout	Im		
Deploy timeout	5m		
General timeout	10m		
Instrument timeout	50		
Network discovery timeout	50 15c		
Remote call timeout	5		
Screenshots on AndroidOS	True		
USB discovery timeout	im		
✓ Mouse			
Mouse Move Interpolation Mode	Optimized		
✓ MSAA			
Evaluate computationally expensive attributes	False		
Filter elements	True	v	
Filter Windows Forms elements	True		
		Restore defaults	
User settings / Solution settings stored on your mach	une 🔟		

Restrictions

- To automate system apps a USB connection has to be established.
- Highlighting on the device does not work for system apps

Validate an Issued Notification

The following example will show how to validate a text in a received notification.

Create a new recording and select the mobile option as usual. Ranorex Spy will open and you will notice a node labeled 'MobileApp AndroidOS' at the application level in the object tree.



To navigate through the element tree you can either use the tree view. Another way to navigate is to use the image navigator which can be found at the bottom of the Overview/Advanced tab. Clicking a UI element selects it, double-clicking outside the selected element selects the parent. To open the notification bar, a swipe action from the top of the screen has to be performed. Navigate to a tree element including the navigation bar in the representing screenshot and add this element to the repository using the context menu.

🗶 Ranorex Spy - (32bit) - Live						_		×
					🖹 SE	TTINGS	1	Studio
TRACK /mobileapp[@title='A	ndroid OS'] /?/?/container [@rid	='R.id	.content']]/?/?/q	:ontainer[@rid=	'R.id.drag_l	ayer']	
BROWSER & RESULTS PATH EDIT	DR							
🖵 C 🕏 💁 🗖		LIVE	Contain	er 'R.i	d.drag_layer'			
MobileDevice 'Nexus 5X'		^	0, 0		1080 x 179 🖬	4		
im MobileApp 'Android OS'			Overview	Advan	ced			
Container 'R.id.content'			General					~
🚊 🛄 Container 'R.id.launcher'			Enabled		True			
Container 'R.id.drag	laver				True			
	Highlight element				True			
	Update element data			er —				
i i i i i i i i i i i i i i i i i i i	Set element as root							
	Add to repository		IE	rType	Frame			
	Add to repository (incl. Children)					Edit path we	aghts	¥
Forr Program	Save as snapshot		1	tainer	'R.id.drag	laver'		-
Container 'R	Capture screenshot)		
					ו=	♥ 0 ¥ 229		
⊡ □ Con	Expand all				Google Say 1	Dk Google' 👃		
	Collapse all							
Container R	Ensure visible							
M Pictu	Focus							
E. Contain	SetOrientation (MobileUiElement)							
Pictu	Touch (MobileUiElement)							
₩ Picture 'R.	id.inactive'							
🖃 🗔 Container 'R.id.ho	tseat'				G 🗊 🤇	>		
	nt 'R.id.layout'					Pay PaySor		
Form and	roid.view.view ement 'android.view.ViewGroup'					9		
		¥						
Tracking OFF: Press Ctrl+LWin for instant tra	icking.							.::

Add a swipe action by dragging/dropping the newly created repository item to the actions table and choosing 'Swipe Action' as action type from the context menu.

ValidateNotification.rxrec* \times					÷
RECORD			🖜 VARIABLES	🖹 SETTINGS) 🌾 Studio
🔁 Add new action 👻 🕅 📋 📋	≣ ।୨ ୯° ▲ ▼ =1≽ Tur	bo mode Screenshot			
# Action	Touch]			
Add new actio	Swipe gesture	ing the Add new action actions into the table	ion	No action s	selected
	Mobile key press Validation Invoke action Get value Set value		>	(52)	
G Add new item ▼ @ Track ▼	Close application] TestSolutionRepository •	Search.	(F3)	P+ 2
AndroidOS	Wait for	itle='Android OS']	Comment	Container	@rid='R.i
RidDragLayer	Capture screenshot Create snapshot User code		[@rid	Google	Say Tak Sangka"
<			>	🔽 📃 🌐	o 💿 💿

In the properties pane set the swipe direction to '90°' and the start location to '0.5;0.0' which is the top center of the element.

ValidateN	lotification.rxrec						Ŧ	Er	dpoints	Properties ×	
R	ECORD	RUN			- VARIABLES	🖹 SETTINGS	🌾 Studio		J₽]	
							_		Duration	ı	500ms
🔁 Add n	iew action 👻 🕅		9 @	🔺 💌 🕸 T	urbo mode 🛛 🔂 Scre	enshot			Enabled		True
#	Action	Swipe dire	Distance	Swipe dura	Repository item			>	lmage b	ased	(None)
" b 1	Swine gesture		500 ×	500ms ~	RidDragLaver				Reposito	ory item	RIdDragLayer
	singe gestare							>	Start loc	ation	.5;.0
						No screensho	t available	>	Steps		0
								>	Swipe d	irection	Down (90°)
								\rightarrow	Swipe d	uration	500ms

Manually open the notification bar on your device and switch back to Ranorex Spy. Navigate through the element tree until you'll find the element you want to validate.



Add the specific element to the repository and add a validation action on the repository item as described before.

Additionally, add a key press on the back button to return to the initial situation.

After adding these three actions, the recording is ready to be executed.

Test Android Wear apps

Following the instructions below you can easily create automated tests on compatible Android Wear apps directly on real or emulated devices.

Prerequisites

• Install adb:

The Android Debug Bridge (adb) is a versatile command line tool that lets you communicate with an emulator instance or connected Android device. You can find it at BinRxEnvAndroidtools.

 Connect with the phone: Download the Android Wear app on your phone and pair the watch with your phone. You can find detailed information on how to do so here: https://support.google.com/androidwear#topic=6056405

• Enable the debug mode:

On the smartphone go to "Developer Options" and enable "USB Debugging". On the watch got to "Developer Options" and enable "ADB Debugging" as well as "Debug over Bluetooth".

• Finally, open the Android Wear app on your smartphone, press the Settings button on the top right and check the "Debugging over Bluetooth". The target should be shown as connected.

🚹 Note

If the target is still disconnected, try to restart both the smartphone and the watch.

Getting Started

Connect the device via USB to your computer and call adb via command line by typing "adb devices".

It should print something like this:

```
List of devices attached 08937p69d0518aac device
```

This means you have successfully connected your phone to your computer. Now we need to connect the watch too. Type the following commands:

```
adb forward tcp:6666 localabstract:/adb-hub
adb connect localhost:6666
```

You need to accept the connection on your watch.

Type once again "adb devices". You should now see a new device called localhost:6666:

```
08937p69d0518aac device
localhost:6666 device
```

In your Android Wear App, Host and Target should now show up as connected.

1 Note Connecting host and target does not always work at first go. Try restarting both devices in different order and be patient.

Now you can connect your devices, instrument your apps, and create your test.

Automate on an Emulated Device

The following instructions explain how to connect a Genymotion emulated smartphone with a Google emulated watch.

First, download Genymotion setup and start any Genymotion emulator with Android 4.3 or later. For Google's emulated watch you need the AVD Manager.

Start the Android Studio, go to Tools -> Android -> AVD Manager. Create a new Virtual Wear device and start it.

Next, install Google Apps or Gapps on the Genymotion smartphone. With Gapps, also the Play Store will be installed, which is needed for downloading the Wear App.

Go to https://opengapps.org/#downloadsection and download the correct Gapps for your emulator. To install Gapps on Genymotion, please follow these instructions.

Gapps may sometimes be unstable when running on Genymotion emulators.

Follow the steps explained in →Prerequisites and list your devices via adb as described in →Getting started.

It should print something like this:

List of devices attached 192.168.154.102:5555 device

This means you have successfully connected your emulated phone to your computer. Now you need to connect the watch too. Type the following command into the command line:

adb -s {genimotion_ip}:{genimotion_port} -d forward tcp:5601 tcp:5601

Type once again the "adb devices" command. You should see a new device called emulator-5600.

Now you can connect your devices, instrument your apps, and create your test.

Instrumentation with source code on iOS

In this chapter, we'll show you how to instrument your app using source code on iOS. Simply follow the instructions below. Alternatively, there are also many video tutorials on YouTube that detail this process.



We recommend you only instrument with source code if you are an experienced iOS developer.

Attention

Whenever you update Ranorex Studio (i.e. use newer automation libraries/Ranorex Service App), **reinstrument your apps**. Otherwise, automation may not work as intended.



Λ

Download the automation lib from the **Mobile Download Archive** on your Mac.

Note

The file size of Ranorex Automation Library is about 30 megabytes. This does not automatically make your app 30 megabytes bigger since it is a universal library and only the parts essential for your app will dynamically be added when compiling it.



Open the XCode project of your application under test.

000	Documentation	
		>>
FAVORITES	Name	A
	lockLandscape~ipad.png	
SHARED	IockPortrait@2x~ipad.png	
DEVICES	IockPortrait~ipad.png	
	MiniKeePass	
	MiniKeePass.xcodeproj	
	pt_BR.lproj	
	README.md	
	ru.lproj	
	Samples	



To avoid shipping an instrumented app to your customers it is recommended to create a separate target for your app under test. You should select the project file and duplicate the existing target.





Rename the newly created target.



በ Note

As XCode does not automatically update all necessary files, you have to rename them manually. You will have to alter the target itself, the targets .plist file (in the workspace as well as in Build Settings -> Packaging), the targets' project name (Build Settings -> Packaging) and the schemes name (Product -> Manage Schemes).



Add the previously downloaded automation lib to your newly created target.

00		MiniKeeD	DY)	Dhone F	1 Cinculat
0.		Minikeer	ASS RA /	chome 5.	1 Simulat
in T	@ A			⊳ I B M	IniKeePass
	iniKeePa	SS SDK 5.1	PROJEC	T.	
		Show in F	inder		
		Open with	Externa	al Editor	
		Open As			•
		Show File	Inspecto	or	
		New File			
		New Proje	ct		
		No. C			
		New Grou	p from (alastian	
		New Grou	p from :	selection	
		Sort by Na Sort by Ty	ame /pe		
		Search in	Selected	Group	2
	. E	Add Files	to [≞] Mini	KeePass"	
		Delete			
		Source Co	ntrol		•
		Project Na	vigator	Help	•

	RxAutomation	; Q
FAVORITES	Name	Date Modified
SHARED	ibRxAutomation.a	Today 12:31 PM
DEVICES		
MEDIA		
Destinatio	on Copy items into destination groups	up's folder (if needed) ders
	 Create folder references for any 	added folders
Add to targe	ts MiniKeePass	August .
Anna Arana	MiniKeePass RX	
New Folder		Cancel Add

6

After doing this, the automation lib will be listed in the 'Linked Binary With Libraries' list in the 'Build Phases' pane of the test target.

PROJECT	Summary Info	Build Settings	Build Phases	Build Rules
MiniKeePass			(Q.	
TARGETS	► Target Dependencies (0	items)		
MiniKeePass MiniKeePass RX	▷ Compile Sources (76 iter	ms)		
	V Link Binary With Librarie	es (8 items)		
	UIKit.framework		Req	uired 🗘
	Foundation.framework	Req	Required \$	
	CoreGraphics.framewo	Req	uired ‡	
	Security.framework	Req	uired 🗘	
	AudioToolbox.framewo	ork	Req	uired ‡
	libxml2.dylib		Req	uired ‡
	libz.dylib		Req	uired 🗘
	libRxAutomation.a		Req	uired 🗘
	+ -	Drag to reorder fra	ameworks	
	► Copy Bundle Resources	(189 items)		



Add the CFNetwork framework to the list.



Additionally, with iOS 8.3 add the IOKit.framework.



In the **Build Settings** pane of the test, add the switches '-ObjC -all_load' to the option **Other Linker Flags**.

Summary Info	Build Settings	Build Phases	Build Rules		
Basic All Comb	ined Levels	Q.*			
Setting		MiniKeePass RX			
▶Kernel Module					
▼Linking					
Bundle Loader					
Compatibility Version					
Current Library Version	1				
Dead Code Stripping		Yes ‡			
Display Mangled Name	5	No \$			
Don't Create Position In	ndependent Exe	No \$			
Don't Dead-Strip Inits	and Terms	No \$			
Dynamic Library Install	Name				
Exported Symbols File					
Initialization Routine					
Link With Standard Lib	raries	Yes ‡			
Mach-O Type		Executable \$			
Order File					
Other Linker Flags		-ObjC -all_load			
		<multiple values=""></multiple>			
Debug		build/MiniKeePass.bui	ld/Debug		



Under the Info tab, add a Bundle display name key (CFBundleDisplayName).

Insert the URL schema in the application settings to open the instrumented IPA from RxServices or Ranorex Studio, for more information review the iOS Service App page. Below, an example of a schema:

```
<key>CFBundleURLTypes</key>
<array>
<dict>
<key>CFBundleTypeRole</key>
<string>Editor</string>
<key>CFBundleURLName</key>
<string>ranorex.your.application.name</string>
<key>CFBundleURLSchemes</key>
<array>
<string>ranorex.your.application.name</string>
</array>
</dict>
</array>
</array>
```

Attention

If a designer implements the existing Apple API, which restricts any schema from running the application, the instrumented IPA does not run within RxServices and Ranorex Studio.



Run the RxServices IPA, put it into the background, and next, **run** the instrumented IPA and put it in the background as well. This makes the application recognizable by the RxServices IPA.

The new IPAs should display in the list of available IPAs in the RxServices application and in Ranorex Studio.

After performing these steps, your project can be built using the newly created target and scheme for your iOS devices as well as simulators.

በ Note

If you see the debug output '**RxAutomationEngine init**' in the console pane, your app has been instrumented successfully.

💌 II 🖙 🛓 🏦 🎢 🎦 MiniKeePass	
All Output \$	
2013-01-03 13:16:55.417 MiniKeePass[739:11603] <info loading="" runtime="" uicontrol+rxautomationcategory="" =""></info>
2013-01-03 13:16:55.452 MiniKeePass[739:11603] <debug1 delegate="" for="" intercepted="" p="" runtime="" set="" textfiel<="" =""></debug1>
of type <minikeepassappdelegate>></minikeepassappdelegate>	
2013-01-03 13:16:55.676 MiniKeePass[739:11603] RxAutomationEngine init
2013-01-03 13:16:55.677 MiniKeePass[739:11603] INFO: Discovery service started on port: 31000
2013-01-03 13:16:55.678 MiniKeePass[739:11603] INFO: Listener socket started on port 31000
2013-01-03 13:16:55.679 MiniKeePass[739:11603] <info <<="" communication="" on="" p="" port="" rxdeviceservice="" started="" =""></info>
2013-01-03 13:16:55.679 MiniKeePass[739:11603] RxAutomationDispatcher init (static)
2013-01-03 13:16:55.680 MiniKeePass[739:11603] INFO: Listener socket started on port 31001

- Because the Ranorex automation lib uses non-public APIs, make sure that you *do not submit a Ranorex instrumented app to the app store* as your app might be rejected and you might be banned from submitting apps to the app store for a period of time.
- To add your device via Wi-Fi, it's necessary that the application under test is started on your iOS device or simulator.
- The development provisioning profile should be used for testing your app and not the distribution provisioning profile.
- To improve the object recognition of your app under test set the accessibility label attribute of your controls.
- Due to restriction in Apple API it may occur that the instrumented IPA is not able to run from with Ranorex Studio, while RxServices application is in the background mode. It is recommended to perform any test while the instrumented IPA is in the foreground.