ArtOfTest Inc.

Automation Design Canvas 2.0 Beta Quick-Start Guide

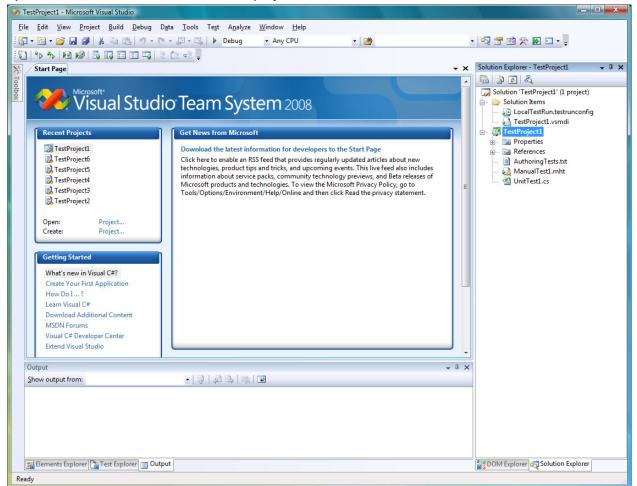
Contents

Creating and Running Your First Test	3
Adding Quick Verification Steps	
Creating Advanced Test Verifications	
Creating a Data Driven Test	
How to resolve test step failures	21
Changing how an Element is found	
Using the 3D Viewer	25
Performing Common Automation Tasks	
Automation Design Canvas Recommend Tool Window Layout	
WebAii Test Run Configuration	
Recording Surface Tool Window Overview	
Element Explorer Tool Window Overview	
Test Explorer Tool Window Overview	
DOM Explorer Tool Window Overview	
WebAii Test Tab Overview	
Customizing Your Test Using C# or VB.NET Code	41
Creating a Test With a Code Behind File	41
Creating a Test With Custom Code Steps	43
How to Reference Elements from the Element Explorer in Code Behind Files	45

Creating and Running Your First Test

Let's jump right in and show you how easy it is to record and run your tests.

1) Open Visual Studio and create a new test project.

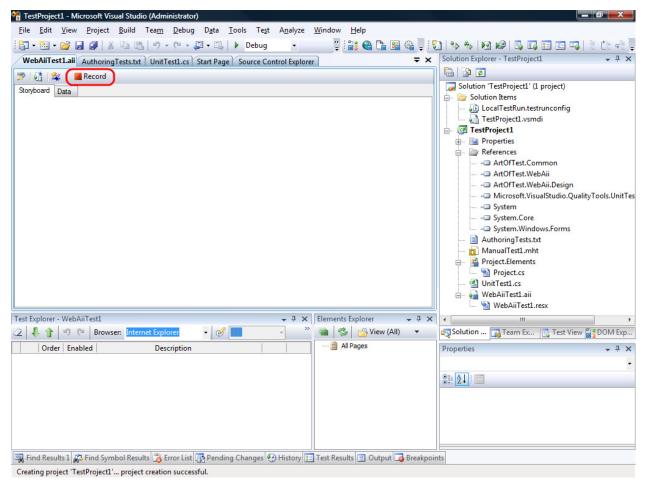


2) Right click on the project node in the solution explorer and select "Add -> New Test...".

Add New Test				? ×
Iemplates:	Generic Test Generic Test Ordered Test	Load Test	Manual Test (text format) Unit Test Wizard	Description Use a WebAii Test to record browser based scenario tests for your web applications. WebAii tests can be used to record actions against web elements, verify web elements' states and content and to also perform test synchronization to help automate rich internet applications (RIAs) that utilize dynamic technologies like Ajax. WebAii tests can be recorded once and then run against InternetExplorer or Firefox.
Test <u>N</u> ame:	WebA	iiTest1.aii		
<u>A</u> dd to Test F	Project: 🔯 Tes	tProject1		*
				OK Cancel

- 3) Click on the "WebAii Test" type and select OK.
- 4) Your new empty test should open automatically.

4



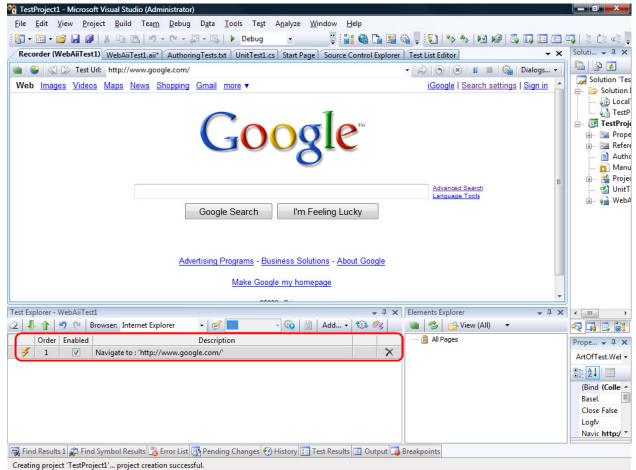
- 5) If this is your first time creating a WebAii Test please see the section "Recommended Tool Window Layout".
- 6) To begin recording the steps of your test, click on the "Record" button (circled in red above).

7) The "Recorder" tab will automatically activate.

ie fait view Project Build Team Debug Data Iools Test Analyze Window Help ■ ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ●	Die Edit Verw Project Build Team Debug Data Iools Test Analyze Window Help Image: State Project Build Test I Authoring Teststat UnitTestLics State Page Source Control Explorer Test List Editor Recorder WebAilTestI Image: State Page Source Control Explorer			_ 0 <u>_ x</u>
Image: Supporer - WebAilTestl Image: Suppore - WebAilTestl Image: Suppore - WebAilTestl Image: Suppore - Image:	Test Explorer - WebAilTest1 Test Explorer - WebAilTest2 Test Explorer - Test Expl			
Recorder (WebAirTest1) WebAirTest1) WebAirTest1) WebAirTest1 Test Ult WebAirTest1 Test Ult Test Ul	Recorder (WebAilTest1) WebAilTest1ai Authoning Tests.tat UnitTestLas Start Page Source Control Explorer Test List Editor Solution Image: Source Control Explorer Test Explorer - WebAilTest1 Image: Source Control Explorer Image: Source Control Explorer Image: Source Control Explorer Image: Source Control Explorer Test Explorer - WebAilTest1 Image: Source Control Explorer Image: Source Control Explorer Image: Source Control Explorer Image: Source Control Explorer Test Explorer - WebAilTest1 Image: Source Control Explorer Image: Source Control Explorer Image: Source Control Explorer Image: Source Control Explorer Image: Source Control Explorer Image: Source Control Explorer Image: Source Control Explorer Image: Source Control Explorer Test Explorer - WebAilTest1 Image: Source Control Explorer Image: Source Control Explorer Image: Source Control Explorer Image: Source Control Explorer Image: Source Control Explorer Image: Source Control Explorer Image: Source Control Explorer Image: Source Control Explorer Image: Source Control Explorer Image: Source Control Explorer Image: Source Control Explorer Image: Source Control Explorer Image: Source Control Explorer Image: Source Control Explorer Image: Source Control Explorer Image: Source Control Explorer Image: Source Control Explorer Imag			
est Explorer - WebAuTest1 t Conder Enabled Description Add C C C C C C C C C C C C C C C C C	Test Explorer - WebAiTest 2] + 🛅 + 🚰 🛃 🗿 X 📭 🖺 17 + 14 + 17 + 18 🕨 Debug 🛛 - 🦉 🔡 🚭 🕞 🛽	8 🍇 - 19 🔊 🎝 🔊 👘 🕅 👘 🖬 🗉 🖬 -	
st Explorer - WebAilTest1 st Explorer - WebAilTest1 Corder Enabled Description Al Pages Prope • All Pages Prope • All Pages Prope • All Pages Prope • All Pages Prope • All Pages Prope • All Pages	Text Explorer - WebAilTest 2 M Browsen: Internet Explorer	lecorder (WebAiiTest1) WebAiiTest1.aii AuthoringTests.txt UnitTest1.cs Start Page Source Control Explor	rer Test List Editor 🗸 🗸 🗸	Soluti 👻 👎
2 1	2 ↓ ↑ ♥ Browser: Internet Explorer ↓			Solution " Solution " Solution " Solution " Compared to the solution Solution " Solution " TestPre Pre Pre Pre Ref Au Au - Au
ArtofTest.V @ii 2010 (Bind (Co Basel	ArtOfTes	🔸 👔 🤊 🥲 Browser: Internet Explorer 🔹 📝 🚾 🕞 🍪 📓 Add 🛛 🖏 🎯	🐚 😒 📴 View (All) 🔻	2 🖪 💷
(Bind (Co Basel	(Bind (Basel Close F LogN			ArtOfTest.Web
Basel	Basel Close F LogN			
Close Fal	LogN			
				Close False
Navic htt	Navic			
	📱 Find Results 1 💭 Find Symbol Results 🔀 Error List 🕞 Pending Changes 🐑 History 🔃 Test Results 🗐 Output 🥃 Breakpoints			Navic http:

8) In the "Test Url" box, enter <u>www.google.com</u>.

9) Click the "Navigate to URL" button.



10) You will notice a recording step has been added to the "Test Explorer" tool window.

- 11) Type in "WebAii" in the Google search box and click the search button.
- 12) You will notice that two more steps have been added to "Test Explorer".

Т	est Exp	olorer - \	WebAiiTes	;t1			-		•	дх
4	2 🦊		🤊 (°	Browser: I	nternet Explorer	- 🧭 400	-	Add	- 🕄 🤇	\$
		Order	Enabled			Description				
	4	1	V	Navigate	to: 'http://www.g	oogle.com/'				\times
	4	2	V	Set 'QTex	d' text to 'WebAii'					×
	4	3	V	Click 'Btn	nGSubmit'					×

That's how easy it is to create your first test! Save and build your project. To execute your test you have 2 options:

- a. You can click the Quick Execute button (circled in red above).
- b. To execute your test using Visual Studios testing framework
 - i. open the "Test View" tool window. This can be found by clicking on "Test" on the top menu then -> "Windows" -> "Test View".

ii. Right click on your test and click "Run Selection".

Tes	st Viev	v	→ ‡	×				
8	I All Columr -							
i	Item	(s) selected: 1						
	Test	Name	Project					
	Manu	ualTest1	TestProject1					
1	Test	/lethod1	TestProject1					
•	Web/	AiiTest1	TestProiect1					
	₽⊳.	Run Selection	on					
		Debug Sele	ction					
		Open Test						
	5] New Test						
	*	Cut						
		Сору						
	B	Paste						
	×	Delete						
•	Select All							
	Add/Remove Columns Open Test List Editor Properties							
DC								

13) An Internet Explorer window will open and the steps of your test will automatically execute in that browser window. When the test completes the browser window will close.

Test Results			
🛅 🚡 📑 🕼 All	•	🏤 Run 👻 🕼 Debug 👻 I	I 💷 🗄 🗗 - 🍋 🤇
 1 test run(s), Result 	ts: 1/1 completed, 1 passed, 0 fai	led Results: 1/1 passed;	
Result	Test Name	Project	Error Message
🗆 🖬 🥥 Passed	WebAiiTest1	TestProject1	Overall Result: Pass
•			
Breakpoints 📰 Test	Results 🔜 Find Results 1 🚜 Fin	id Symbol Results 🧐 Histor	y 📸 Error List 📑 Pendi
Ready			

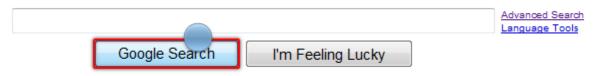
14) After the test runs, double click on results summary in the "Test Results" tool window to view the execution log.

WebAiiTest1 [Result	: 1:40 PM] Recorde	r WebAiiTest1.aii	AuthoringTests.txt
- Common Results			
Test Run: Test Name: Result: Duration: Computer Nam Start Time: End Time: — Error Message	@AOT- WebAiiTest1 @ Passed 00:00:03.1282783 e: AOT- 10/27/2009 1:40:49 10/27/2009 1:40:52		0:48
Overall Result: Pa	ISS		
'10/27/2009 1:40:	51 PM' - 'Pass' : 1. Na 51 PM' - 'Pass' : 2. Se 52 PM' - 'Pass' : 3. Cl	t 'QText' text to 'We	
'10/27/2009 1:40:	52 PM' - Overall Resu	ult: Pass	

Adding Quick Verification Steps

Adding verification steps is quick and simple using the Quick Tasks menu.

- 1. In the Recorder window hover the mouse cursor over an element you would like to add a verification on.
- 2. Wait 1 second for the little blue round nub to appear.



3. Click on the nub. This will open the Element Toolbox.



4. Click on the Quick Tasks button (circled in red above). This will open a list of available quick tasks. What is listed depends on what type of element you have selected. Here's an example with a select drop down selected:

English	Quick Tasks HtmlSelect	
Translate and §	Prify - selection 'Text' is 'English'	
<u>d search</u> to restrict y	Verify - selection 'Value' is 'en'	11
a search to restrict y	🅞 Verify - element is visible	ш
t	🅞 Verify - element is not visible	88
	Wait - selection 'Text' is 'English'	
	Wait - selection 'Value' is 'en'	
	🔊 Wait - element exists	
	💩 Wait - element does not exist	$\overline{\Delta}$

Notice that is has "Verify selection is" items.

5. Here's an example with a Table element selected:



Notice that it has "Verify table has" items.

6. Double clicking one of these items will add that Verify/Wait For as a step to your test list this:

5 1		🔊 (°	Browser: Internet Explorer 🔹 🥩 🔟 🕞 Add.	🔞 🔇
	Order	Enabled	Description	
	5	V	Verify table's 'Verification' is '1' on 'Table'	×
	6	V	Verify table's 'Verification' is '3' on 'Table'	X
	7	V	Verify table's 'Verification' is '2' on 'Table'	X
	8	V	Verify element 'Table' 'is' visible.	×
1	9	V	Verify selection 'ByValue' is 'en' on 'SISelect'	×

Creating Advanced Test Verifications

🤣 TestProject1 - Mi	crosoft Visual S	tudio				- I Barnati	the Second	Theorema in .		
<u>File Edit V</u> iew	<u>P</u> roject <u>B</u> ui	ld <u>D</u> ebug	D <u>a</u> ta <u>T</u> o	ools Te <u>s</u> t	A <u>n</u> alyze	<u>W</u> indow <u>H</u> elp				
i 🛅 + 🛅 + 💕 🕻	0 X D	11 19 -	(H - 19	- 🖪 🕩	Debug	 Any CPU 		2		1
		E E 🖏								
	aii.cs WebAiiT				NobAiiTorti	•				- × ◙
WebAllTest1.										00 D
WebAiiTest1.	est Uri: http://	www.google	.com/searc	n:ni=en@c	l= MepAlloc	tnG=Google+Searc	n	• 0 9		
X										xpl
				Sentenc	e Verificat	ion Builder	0			orer
										2
Select a	sentence and pi	ck its values								Solu
										Solution Explorer
	nent: 'inputl' (nam								e.a 🗸	Expl
Attr	ibutes 👻	🕹 name	Exact	q Edit				~ _	8 ×	orer
	Validation	Passed!								
										TestView
[Eler	nent 'inputl' (nam	ne=q)]								VIEV
	•							V -		
										J
										Properties
										Dertie
										0
								ОК	Canc	el
							L			
Elements Explore		X Test E				6.97		Les L		→ ₽ ×
👜 📴 View		2.	🖡 🚹 🛛 Br	rowser: Int	ernet Explor	er 🔹 📝 👖	•	🥪 📓 Ad	d 🔻	
E 🗐 No Active			Order	Enabled			cription			
				V	-	o : 'http://www.goo	-			X
	input_Text_Q	- 4		V		Text_Q' text to 'Web	Aii'			×
····· =	input_Submit_E					It_Submit_BtnG'	Coded C			×
		<u> </u>	\$ 4	V	[MyCusto	mStep] : My Custom	Coded Step		85	×
		-		-						
		Te	st Explorer	Output	SE Test Res	ults				
Ready										

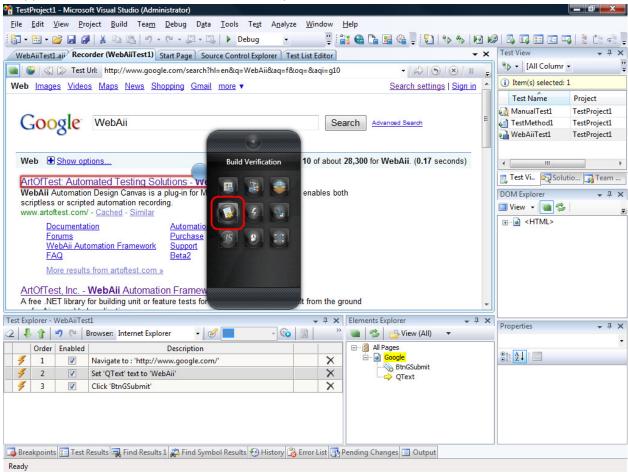
- 1) Test Verifications are added to a test as an ordered step.
- 2) To craft a verification step, enable the Automation Overlay Surface by clicking on the "Enable overlay" button.



3) You will know if the Automation Overlay Surface is on by hovering over the web page and seeing each element being highlighted.

😭 TestProject1 - Microsoft Visual Studio (Administrator)	
<u>File Edit Vi</u> ew <u>P</u> roject <u>B</u> uild Tea <u>m D</u> ebug D <u>a</u> ta <u>T</u> ools Te <u>s</u> t A <u>n</u> alyze <u>W</u> indow <u>H</u> elp	
🞼 = 🔤 = 😂 🔒 🎒 🔏 🖻 🛍 🗉 - 🔍 = 💷 🕨 Debug 💿 = 🕎 🕌 🍓 🔓 🖼 🍓 📮 🖏 🍫 🆘 🗺 K	9 5, 15 🗉 🗉 📭 🛓 🖄 🧔 📮
WebAiiTest1.aii Recorder (WebAiiTest1) Start Page Source Control Explorer Test List Editor + ×	Test View → ╀ 🗙
💽 🥪 🔍 🗁 Test Url: http://www.google.com/search?hl=en&q=WebAii&aq=f&oq=&aqi=g10 🛛 🔸 🎣 🕤 🛞 💷 ᆯ	P≩⊳ - [All Columr - 🚆
Web Images Videos Maps News Shopping Gmail more - Search settings Sign in	(i) Item(s) selected: 1
	Test Name Project
	ManualTest1 TestProject1
Google WebAii Search Advanced Search	TestMethod1 TestProject1
Ũ	WebAiiTest1 TestProject1
Web E Show options Results 1 - 10 of about 28,300 for WebAii. (0.17 seconds)	
	Test Vi Solutio Team
ArtOfTest: Automated Testing Solutions - WebAii Automation	
WebAii Automation Design Canvas is a plug-in for Microsoft Visual Studio; that enables both scriptless or scripted automation recording.	DOM Explorer - 4 ×
www.artoftest.com/ - <u>Cached</u> - <u>Similar</u>	🔲 View 👻 📳 😴
Documentation Automation Design Canvas	⊞ 🗃 <html></html>
Forums Purchase WebAii Automation Framework Support	
FAQ Beta2	
More results from artoftest.com »	
ArtOFFact Inc. Web All Automation Framework	
ArtOfTest, Inc WebAii Automation Framework A free .NET library for building unit or feature tests for rich web applications. Built from the ground	
Test Explorer - WebAiiTest1 → ┦ × Elements Explorer → ┦ × Q ↓ ↑ ♥ ● ● ● ● ● ● ✓	Properties 👻 🕂 🗙
	•
Order Enabled Description ✓ 1 ✓ Navigate to : 'http://www.google.com/'	
BtnGSubmit	
✓ ✓ ✓ ✓ ✓ 3 ✓ Click 'BtnGSubmit' ×	
🗔 Breakpoints 🗊 Test Results 🖼 Find Results 1 🕋 Find Symbol Results 🧐 History 👸 Error List 🚯 Pending Changes 🗐 Output	32
Ready	

4) Select an Element that you want to craft a verification for, wait 1 second for the little blue nub to appear, and then click on the nub. This will open the Element Toolbox.



5) From the Element Toolbox, click the "Build Verification" button to open the "Sentence Verification Builder".

😭 TestProject1 - Microsoft Visual Studio (Administrator)	
<u> </u>	
🖥 • 🔤 • 😂 🛃 🥔 🙏 🖻 🖭 🔊 • 🔍 • 📮 • 🖳 🕨 bebug 🔹 🔮 🚼 😫 🍓 📮 🖏 🕪 🆘 👀 🖄	9 6 6 6 6 6 6
WebAiiTest1.aii Recorder (WebAiiTest1) Start Page Source Control Explorer Test List Editor	Test View 🗸 म 🗙
💽 🎯 🔍 ▷ Test Url: http://www.google.com/search?hl=en&q=WebAii&aq=f&oq=&aqi=g10 🔹 🖌 🕤 🛞 💷 👳	P> ▼ [All Columr ▼
Sentence Verification Builder OK Cancel	(i) Item(s) selected: 1
	Test Name Project
	ManualTest1 TestProject1
Select a sentence and pick its values	TestMethod1 TestProject1
	WebAiiTest1 TestProject1
	4
	Test Vi 🟹 Solutio 🏹 Team
	DOM Explorer 🗸 🕂 🗙
	🔲 View 👻 🔚 🤣 🚽
	HTML>
Test Explorer - WebAiiTest1 + 7 × Elements Explorer + 7 ×	
Test Explorer - WebAiiTest1	Properties 🗸 🕂 🗙
	•
Order Enabled Description ∮ 1 ☑ Navigate to : 'http://www.google.com/'	3∎ 2 ↓
BtnGSubmit	
✓ ✓ ✓ ✓ ✓ 3 ✓ Click 'BtnGSubmit' ×	
📴 Breakpoints 😰 Test Results 🖼 Find Results 1 🟩 Find Symbol Results 😏 History 📸 Error List 🚯 Pending Changes 📃 Output	
Ready	

- 6) The verification builder offers sentence based verifications of elements.
- 7) Start by selecting a rule category from the drop down box.
- 8) During verification crafting, the content is dynamically built against the currently selected element. As you make choices, default values will be populated according to the values the element contains.
- 9) For example, if you choose "Content" as your verification rule then you will see three menu options. Click on the down arrow next to each option to see a list of possible values.



🏫 TestProject1 - Microsoft Visual Studio (Administrator)	
File Edit View Project Build Team Debug Data Tools Test Analyze Window Help	
🛅 = 🖼 = 💕 🖬 🍘 🌡 🐁 🛍 🗉 - 🗠 - 📮 = 🖳 🕨 Debug 🛛 - 🙄 🕌 🚭 🕞 🔚 🍓 📮 🖏 🆘 🆘 🕅 🖉	
WebAiiTest1.aii/ Recorder (WebAiiTest1) Start Page Source Control Explorer Test List Editor - ×	Test View 🗸 🕂 🗙
💽 🚭 🔇 ▷ Test Uri: http://www.google.com/search?hl=en&q=WebAii&aq=f&oq=&aqi=g10 🔹 + 🌧 🕤 🛞 🗉 🍦	Print and the second s
Sentence Verification Builder OK Cancel	(i) Item(s) selected: 1
	Test Name Project
	ManualTest1 TestProject1
Select a sentence and pick its values	TestMethod1 TestProject1
	WebAiiTest1 TestProject1
Content Content	
Verify content? <u>compare?</u> value?	4
InnerMarkup OuterMarkup	Test Vi 🔁 Solutio 📷 Team
TextContent	DOM Explorer 🗸 🕂 🗙
StartTagContent	🔲 View 👻 💼 🤣 🖕
Test Explorer - WebAiiTest1 - + A × Elements Explorer - + + ×	
📿 🖡 🛉 🧐 😢 Browser: Internet Explorer 🔹 🥩 🔟 🔹 🍪 📄 👋 🚔 View (All) 🔹	Properties - I ×
Order Enabled Description Grad Pages	
Image: Second	
🔰 2 🔽 Set 'QText' text to 'WebAii' 🗙	
🔰 3 🐨 Click 'BtnGSubmit'	
🗔 Breakpoints 📴 Test Results 🖼 Find Results 1 🥋 Find Symbol Results 🤣 History 蹪 Error List 📑 Pending Changes 🗐 Output	
Ready	

- 10) You can validate that your verification is valid by clicking on the "Verification" icon.
- 11) You can also locate the current element in the DOM tree by clicking the "Locate in DOM" icon.
- 12) To delete or start over, click the "Delete" icon.
- 13) Verifications can be crafted to verify many different values, styles or attributes of an element.You can craft multiple sentences by selecting a rule and filling in the verification criteria. NOTE:Each sentence will add a separate verification step to Test Explorer.
- 14) When you finish building your verification, select OK to add it as a step to your current test.

Creating a Data Driven Test

🔒 Test	Project1	L - Microso	oft Visual Studio (Adr	ministrator)								x
<u>F</u> ile	<u>E</u> dit	<u>V</u> iew <u>P</u> ro	oject <u>B</u> uild Tea <u>m</u>	<u>D</u> ebug D <u>a</u> ta	<u>T</u> ools Te <u>s</u> t A <u>n</u>	alyze <u>W</u> in	dow <u>H</u> elp					
🛐 • 🖼 • 📂 🖬 🆓 X 🖎 🖄 9 • 9 • 3 • 3 > Debug 💿 💿 🕎 🕌 🚭 🕼 📾 🆓 🚽 🖏 > N 🍫 N 🚱 🖪 🖬 🖏 2 🗈 4 = 4												
Web	AiiTest	1.aii Web	AiiTest1.aii.cs Reco	order (WebAiiTest	1) Start Page Sou	rce Control E					- ×	_
3		Rec										🛄 Test View 🗠 Solution Explorer 📷 Team Explorer
Storyb		ata										st Vie
Colum			6								0	We
		Animals	Location	Season	Expected Count							2 S
1	Giraf		Africa	Summer	86							oluti
▶2	Lion	2010	Asia	Spring	56							I S
*3												xpld
												orer
												5
												Tea
												E
												Xp
												orer
												DOM Explorer
												DON
												plo
												Ē
									er and			
		WebAiiTes	ti Browser: Internet Exp		<i>A</i> 100	es la L	Add 🔀 🚳		Elements Explorer	D -	- ₽ Х	P
2	-		srowser: Internet Exp	100		₩ ○ 🖾 1	Add +	<u>۶</u>		I) •		Properties
		Enabled			Description				E 🗐 All Pages			ties
4	1		Navigate to : 'http:/		n/'			×	BtnGSubmi	t		
4	2		Set 'QText' text to 'V	VebAii				X	QText			
4	3		Click 'BtnGSubmit'					×				
4\$	4	V	[MyCustomStep] : (©™y Custom Co	ded Step			×				
📕 Brea	akpoints	s 😑 Test I	Results 🔜 Find Resu	lts 1 💦 Find Sym	bol Results 🧐 Histor	ry 遺 Error L	ist 📑 Pending Cl	hanges 📃 Out	put			
Itom(c)	Carried											

- 1) Create a new test as outlined above.
- 2) Record the Google search steps as outlined above.
- 3) Click on the test tab.
- 4) There are two tabs at the top of this tab, one labeled "Steps" and the other labeled "Data"
- 5) Click on the "Data" tab.
- 6) At the top of this tab are 3 buttons. Only the "Create a new data table" will be enabled. Click this button to add a new grid for your data.

🖻 🛃 📕 Record						
Storyboard	Data					
Columns:	4	📚 🔒 🔒				
Animals						

7) The default grid will have 5 columns. For this data test we are going to go through five iterations of the test with different search text each iteration.

8) Change the columns text box to 1 and click "Update".



- 9) You now have a grid of just one column.
- 10) Enter any text into the first grid cell and hit enter or tab. The input will move to the second grid cell.
- 11) Continue entering text for 4 more grid cells. New rows will automatically be added as you type.
- 12) Save your test.
- 13) You can use data from a data array in both recorded steps and code behind methods. To use reference data from the data array in a code behind method continue with the following steps. To bind data from a data array to a recorded step, jump to step 18.
- 14) Convert the step to code that sets the value of the Google search text box from the "Test Explorer".
- 15) To reference the data from your grid use the "Data" property followed by the index of the column. For example:

```
[CodedStep(@"Set 'QText' text to 'WebAii'")]
public void WebAiiTest1_CodedStep()
{
    // You can reference the column by index
    Pages.Google.QText.Text = (string)Data[0];
    // Or by name
    Pages.Google.QText.Text = (string)Data["Animals"];
}
```

- 16) Save and build your project.
- 17) Execute your test. Note that the test will execute for each row in the data array.
- 18) To bind data to a recorded step, follow these steps.
- 19) Open the properties window by pressing the F4 key.
- 20) Click on the recorded step that sets the value of the Google search text box from the "Test Explorer".

21) The properties for this step will appear in the properties window.

Properties 🛛								
ArtOfTest.WebAii.Design.IntrinsicTransla 🗸								
	(Bindings)	(Collection) 💌						
	ClosesBrowser	False						
	IsPassword	False						
	LogMessageOnFailure							
	Pause	None						
	PrimaryTarget	/Google/QText						
	RunsAgainst	AllBrowsers						
	SecondaryTarget							
	SimulateRealTyping	False						
	StepType	Action						
	Text	WebAii						
	TypingSpeed	150						
	WaitOnElements	True						
	WaitOnElementsTimeo	5000						

- 22) Click on the drop down arrow for (Bindings).
- 23) Click on the "Text" node in the displayed tree.
- 24) Enter the name of the column you want to draw data from into the text box. In our example we want the first column so we'll enter \$(Animals).
- 25) Click the Set button



- 26) The data for the column named "Animals" from the data array is now bound to the Text property for that step. Instead of entering the text "WebAii" into the search box, the data stored in the data array will be entered instead.
- 27) Save and build your project.
- 28) Execute your test. Note that the test will execute for each row in the data array.

How to resolve test step failures

1) After running quick execution for a test, any failed steps will appear in error:

Τe	Test Explorer - WebAiiTest1							
4	2	dd 🏾 🕄	۵۶ 🔇					
	(3)							
Γ		Order	Enabled	Description				
Γ	4	1	V	Navigate to : 'http://www.google.com/'	×			
[4	2	V	Set 'QText' text to 'WebAii'	×	0		
📕 🥖 3 📝 Click 'BtnGSubmit'			×					
4 📝 Verify 'TextContent' 'Contains' 'ArtOfTest: Automated Testing Soluti.			×	0				

2) Clicking the Red X icon will launch the failure debug UI.



3) The failure debug UI will give a description of the failure as well as details about the failed step.

	and the second s	ect1 - Microsoft	Contraction of the local division of the loc	a de casa de c		-						X
Eil		view <u>P</u> roje • ≌ ⊌ ₽			_				-			
		AiiTest1.aii Sta						-			- La -	
Data		🍹 🔍 💭 Tes										
	-	Step Failure Deta								OK	Cance	solution Explorer
Sources												Explo
		lure Details — ntent does not n	natch									Ter
	Exr	ected: Content:	{Verify Te	xtConter	nt Contain	s ArtOf	Test2 }					
	Par	ameters: SegmentType					ಂದರ್ಶನೆ					ame
	Cor	mpareType = pectedString = I	'Contains'		21 Actus	0-t-Al-	Tost					xpior
		ecteusuing – i	:xpecteu.[A	litorrest	zj, Actua	in[AitO	restj					er
											<u>Copy</u>	
	Res	olve Failure: (Ve	rification Ru	ule) —								Test view
		Select a sent	ence and p	ick its va	lues							DOM
			No live ele	ment								Т
		Content	Verify	TextC	ontent	i Co	ntains	🖡 Art	0fTest2	4		Explorer
											J	
		Note									_	J. PTO
		Live validation	n is disabled.	i.								Properties
												es
										-		
-		t Results 🔜 Fin	d Results 1	🔁 Find	Symbol R	esults	Histor	y 🖪 P	ending C	hanges 違	Error List	
Iter	m(s) Sav	ea					_					_

- 4) The failure debug UI shows you the details about the failure and offers you the chance to update the verification if that is what is needed.
- 5) At this point the tester needs to decide if the test needs updating or the page under test has a defect. If the test needs updating, then you can simply modify the verification properties and click OK to update the test. Otherwise the test has detected a product defect and a bug should be logged in your bug tracking system.

Changing how an Element is found

estProject1 - Microsoft Vi	sual Studio (Administrator)	
	<u>B</u> uild Tea <u>m</u> <u>D</u> ebug D <u>a</u> ta <u>T</u> ools Te <u>s</u> t A <u>n</u> alyze <u>W</u> indow <u>H</u> elp	
	6 🗈 🖺 🗉 - 🔍 - 💭 - 🖳 🕨 Debug 🛛 - 🛛 🦉 🔡 🍓 🖕 🖏 🕪 🖗 M 🕨	
bAiiTest1.aji Recorder	r (WebAiiTest1) Start Page Source Control Explorer Test List Editor 🗸 🗸 🗙	
🍟 🔇 🔝 Test Url:	http://www.google.com/ • 🎝 🔊 🛞 🗉 💂	All Columr -
Find Expression Builder	OK Cancel	(i) Item(s) selected: 1
		Test Name Project
		ManualTest1 TestProject1
Find Expression		TestMethod1 TestProject1
	🗙 name Exact 🔹 btnG AND 🔹 🖶	WebAiiTest1 TestProject1
	X tagname Exact Input AND I	
		Test Vi 🔊 Solutio 🏹 Team
	Add :	DOM Explorer 🚽 🕂
	Exact	🔲 View 👻 👜 🤹
		D width="25%"> D noWrap align=middle>
		<pre><input nam<="" pre="" type="hidden" value="en"/></pre>
Expression		(INDUT calve he take hidden as
name=btnG,tagname	e=input	<input class="lst" google="" se<br="" title="Google Sear-</td></tr><tr><td>Attributes List:</td><td></td><td>SUKA</td></tr><tr><td></td><td></td><td><INPUT class=lsb value="/>

- 1) Whenever a web page element has an action recorded against it, or you explicitly add an element to Elements Explorer, a "Find Expression" is generated that tells the framework how to find that specific element on the web page.
- 2) In order to change how an element is found, right click on the element in the Element Explorer tool window and select "Edit Element".
- 3) The "Find Expression Builder" will launch and display how the element is currently being found.
- 4) The "Find Expression Builder" consists of the following components:
 - a. Element Property/Attribute The first textbox specifies what property or attribute of the element to examine. Some of the things you can enter for HTML elements include: TextContent, NodeIndexPath, TagName, TagIndex, XPath or any HTML attribute (e.g. name, id, visibility, etc.). Some of the things you can enter For XAML elements include: TextContent, XamITag, Name, TagIndex or any XAML attribute (e.g. Foreground, Content, etc.)
 - b. Compare type The compare type drop down controls what sort of comparison will be performed (e.g. Exact, Contains, StartsWith, etc.).

- c. Value to look for The next textbox holds the value that will be used in that expression. It can be either a value to look for (such as a string, or a number) or an index path (tag index or XAML index), or a RegEx expression.
- d. And/Then Find Expression Builder supports the ability to use multiple criteria for locating the correct element. When this drop down is sent to AND it means the current line and the next line must both equate to true. When sent to THEN it means you are creating a chained find expression. In a chained find expression you are specifying how to find element 'A' and then underneath element 'A' find element 'B'. This is very useful for pages in which multiple custom complex controls have been added in different panels. You can use a chained find expression to locate the correct panel and then find the correct element (such as a button) within that panel.
- 5) After specifying the method and criteria to use for the element find, you can click on the "Validate" icon inside the "Actions" group box to validate that the current find expression finds the element within the current DOM.

Actions				_
Element: [Element: 'em:0']	Ι	9	\checkmark	
🔍 🖉 Validation succeeded!				

NOTE: For more information and a detailed discussion on crafting and using find expressions, please visit our <u>website</u>.

Using the 3D Viewer

The 3D viewer shows a hierarchical view starting at a selected element traversing up the DOM tree all the way up to the DOM root. It can be used to identify/lock on elements and quickly build verifications. This is what it looks like:

🦓 TestProject1 - Microsoft Visual Studio (Administrator)		
File Edit View Project Build Team Debug Data Iools Test Analyze Window Help		
📴 🕶 🕶 🚰 🖉 🖉 🖉 👘 🖏 🖏 👘 🕫 - 🖏 - 🖏 - 🖏 - 👘 Debug - Any CPU - 🔹 💆 👘	🍤 🏠 😥 😰 I 🖏 🖬 🖬 📼 🖓 🖄 🖄 🖏 📜	
WebAiiTest1.aii Build Explorer - Products Start Page Source Control Explorer Test List Editor Recorder (WebAiiTest1)	DOM Explorer 🚽 🗸 📑	
📓 🗑 🔇 💭 Tet Url: http://www.google.com/	🔲 View 🔹 🚘 🤣 Page (Google) 🔸 🗥 😨	
Coog ment integing	View Page (Google) Page (Fight) Page (Fi	
Find Element Available Verifications View: Selected Only Add to Project		
<input class="lsb" name="btnG" type="submit" value="Google Search"/>	<input typ<br="" value="en"/> <pre> <input pre="" typ<="" value="hp"/></pre>	
<td align="middle" nowrap=""></td> <td><pre> <input class="lst" td="" title<=""/></pre></td>		<pre> <input class="lst" td="" title<=""/></pre>
<tr valign="top"></tr>	<input class="lsb" vali<br=""/> <input class="lsb" p="" vali<=""/> # 4 CTD width="25%" noWra	
<tbody></tbody>		
<table cellpadding="0" cellspacing="0"></table>	- E - A 	
FOON asmalf action (assets	< >	
	Soluti 📷 Team 🗒 Test V 🔡 DOM	
📮 Breakpoints 😰 Test Results 🖏 Find Results 1 🤹 Find Symbol Results 😔 History 选 Error List 🚯 Pending Changes 🗉 Output 🛛 📓 Elements Explorer		
Ready		

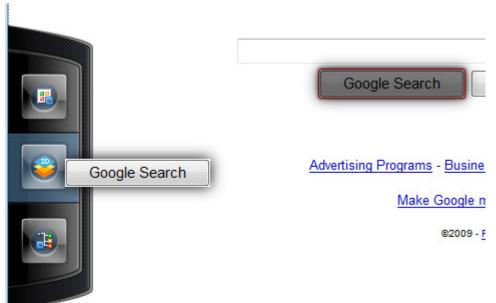
There are two methods of opening the 3D viewer.

- 7. In the Recorder window hover the mouse cursor over an element you would like to examine in the 3D viewer.
- 8. Wait 1 second for the little blue round nub to appear.



9. Start dragging this nub. The nub will change into an image of the highlighted element and a pop out panel will appear on the left side of the screen with three buttons in it. Drop the image of





the element you are dragging onto the middle button (the 3D viewer button)

- 10. The 3D viewer will open and the element you were dragging will be selected in the viewer.
- 11. If you drag the element onto the top button it adds that element to the Elements Explorer tool window.
- 12. If you drag the element onto the bottom button it locates that element in the DOM tree and highlights it in the DOM Explorer tool window.

Once the 3D viewer is open you can:

- 1. Traverse the DOM layers by:
 - a. Clicking on an element in the element list.
 - b. Dragging the blue nub left/right (shown below).

	Lock on Surface		
	Google Search I'm Feeling Lucky	Advantad Search Lanasaan Toob	ana ka ana ka ana ka
C.	Clicking on the left/right arrows (shown above).	•	

- 2. Increase/decrease magnification of the screenshot by clicking the + or symbol (shown above).
- 3. Clicking on the "Lock on Surface" button will:

- a. Close the 3D viewer returning you to the Recorder window.
- b. Highlight that element in the Recorder.
- c. Open the Element Toolbox.
- 4. When you click on the "Available Verifications" tab a list of prebuilt verifications is displayed.

% TestProject1 - Microsoft Visual Studio (Administrator)		_ 🗆 🗙
<u>File Edit View Project Build Team D</u> ebug D <u>a</u> ta <u>T</u> ools Te <u>s</u> t A <u>n</u> alyze <u>W</u> indow <u>H</u> elp		
🛐 🕶 🖼 🕶 🛃 🥔 🐰 🖻 🛍 🔊 + 🔍 + 💷 + 🖳 🕨 Debug 🔹 🖓	: 🔐 😋 🔚 🤀 📮 : 🐑 🆘 🆘 🖻 🔎 🖏 🖬 📰 📼 🧊	
Test Url: http://www.google.com/	• 💫 🕤 🛞 💷 🔳 🆓 Dialogs •	Sol
aso		ution
WebAiiTest1.aii* AuthoringTests.txt UnitTest1.cs Start Page Recorder (WebAiiTest1)) Test List Editor Image: Second start Page Image: Second start Page		×
		lore
	And both the first	
	2	Tear
	Name of the second seco	nEx
		plore
		Tes
		tVie
Find Element Available Verifications View: All Categories V	Selected Only Add to Pro	ject
		X Solution Explored Team Explored Test View
Table Number of RowsCount Equals 1		
🗹 👸 🗹 Table Number of ColumnsCount 💌 Equals 💌 3 🖉		Properties
Content Verify InnerText Contains Advanced		ertie
	Searchenbsp, earbsp, Language 100is	
Test Explorer - WebAiiTest1		→ ₽ ×
2 1 a construction of the second seco	Ø4	• + A
Order Enabled Description		•
🛛 🐔 2 📝 Set 'QText' text to 'ArtOfTest'		X
		×
4 Verify 'TextContent' 'Contains' 'ArtOfTest' on 'ArtOfTestEmTag'		XE
5 Verify table's 'Verification' is '1' on 'Table' 9 6 Verify table's 'Verification' is '3' on 'Table'		×
		· ·
📰 Test Results 🖼 Find Results 1 🔊 Find Symbol Results 🚱 History 📑 Pending Changes 📸 Error List 📃 O	utput 📮 Breakpoints 🛛 🔡 Elements Explorer	
Ready		194

- 5. The controls in the View group box control which of the available verifications will be displayed. You can change the category being displayed using the Categories drop down. You can limit the view to only the verifications you have selected by checking the "Selected Only" checkbox.
- 6. You can quickly add verifications by checking the checkbox (or checkboxes) for the verification(s) that you want and then clicking Add to Project.
- 7. There are two buttons on each verification in the list. The first button (the one on the left) will test whether or not the verification passes or fails against the currently loaded web page. The other button will locate the selected element in the DOM tree and highlight it in the Dom Explorer tool window.
- 8. You also have the option of modifying the verification before you add it to the project. Use the drop downs and the edit boxes to modify the verification before clicking "Add to Project".

Performing Common Automation Tasks

This picture shows an exploded view of the Element Toolbox with all of the different tasks you can accomplish with it:



Automation Design Canvas Recommend Tool Window Layout

The first time you load Automation Design Canvas, all tool windows except for the recording surface might be floating. Below is a recommended layout.

Project1 - Microsoft Visual Studio (Administrator)				
<u>File Edit View Project Build Team Debug Data Tools</u>	Test Analyze Window Help			
- - - -	ebug - Any CPU -	2 🔐 🔒 🐚	🖴 🍇 📮 🐑 🔹 🍫 💌 🖄	
		• · · · · · · · · · · · · · · · · · · ·		DOLLE 1
Recorder (WebAiiTest1) WebAiiTest1.aii Statt Page Test List		-1010	🛛 🛞 🛯 🔳 🍪 🛛 Dial	DOM Explorer ↓ ↓ × ↓ View ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓
Web Line Man Man Obering Oracity		a second and a second s		en i <html></html>
Web Videos Maps News Shopping Gmail m	<u>ne</u> .	<u>IG00</u>	gle <u>Search settings</u> <u>Sign in</u>	
e s				BODY onload="doc
	oogle			
				IIII id=gbar> IIII id=guser wi
	0			<pre> </pre>

The layout above is the recommended tool window layout to optimize space and recording efficiency. The Automation Design Canvas Tool Windows are:

- a. Recording Surface (Document window docked)
- b. Elements Explorer (Dockable tool window docked)
- c. Test Explorer (Dockable tool window docked)
- d. DOM Explorer (Dockable tool window docked)

Note: This is only a recommended layout; you should customize your layout to your liking.

WebAii Test Run Configuration

To configure WebAii Framework settings for execution:

- 1) Double click on "localtestrun.testrunconfig" under "Solution Items" in the "Solution Explorer".
- 2) On the left side of the Test Run Config dialog click on the item labeled "WebAii Test".
- This will give you a page with three tabs to configure all the run time configuration settings for all test executions within this project.

localtestrun.testrunconfig		8 x
localtestrun.testrunconfig General Controller and Agent Code Coverage Deployment Hosts Setup and Cleanup Scripts Test Timeouts Web Test WebAii Test	General Execution File Paths General Settings Default Browser: Local Web Server Type: Client Ready Timeout (msec): Execute Command Timeout (msec): Mouse Move Speed (0-1):	? × InternetExplorer ▼ None ▼ 30000 ↓ 20000 ↓ 0.3 ↓
	Wait Check Interval (msec): WebAii Test run settings. Configure the e	500

Recording Surface Tool Window Overview

Recorder (WebAiiTest1) WebAiiTest1.aii Start Page Test List Editor Source Control Explorer	- ×
💼 🚱 <<>> Test Url: http://www.google.com/	🗸 🌧 🕥 🗷 📕 🛯 🍓 Dialogs 🗸
Web Images Videos Maps News Shopping Gmail more T	iGoogle Search settings Sign in
Google	Advanced Search
Google Search I'm Feeling Lucky	Language Tools
Advertising Programs - Business Solutions - About Google	
Make Google my homepage	
©2009 - <u>Privacy</u>	
	-

The "Recording Surface" tool window contains a web browser and the Automation Overlay Surface. Most actions performed against a webpage can be recorded by just performing the action directly. The menu bar for the Recording Surface contains (from left to right)

- a. Enable / Disable the Automation Overlay Surface
- b. Launch external IE recording window
- c. Navigate Back
- d. Navigate Forward
- e. Test Url
- f. Navigate To
- g. Refresh page
- h. Stop navigation
- i. Pause recording
- j. Start recording (Enable the capturing of actions on page)
- k. User Settings (configure settings for Automation Design Canvas)
- I. Add dialogs drop down

The Automation Overlay Surface (when enabled) provides the user with a visual effect indicating the element you are hovering over.

In order to perform more specific actions or to craft verifications against an Element, hover over an element for one second and a little blue circle will appear that is called the "nub".

Web	Images	<u>Videos</u>	<u>Maps</u>	News	Shopping	<u>Gmail</u>	more V	iGoogle Search settings Sign in	*
						(Google	Advanted Security	
					_			Advanced Search Language Tools	
						Goog	le Search I'm Feeling Lucky		
					Adv	ertising	Programs - Business Solutions - About Google		
							Make Google my homepage		
							©2009 - Privacy		
									-

Clicking on the nub will open the Element Toolbox



The Element Toolbox has these buttons:

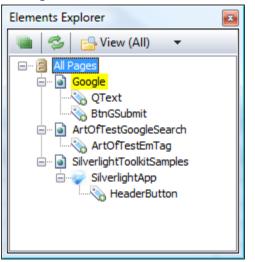
- a. Add to project elements Adds the highlighted element to the Elements Explorer tool window.
- b. Locate In DOM Explorer Highlights the current element in the DOM Explorer tool window.
- c. View 3D Opens a special hierarchical 3D view showing the path from the highlighted element all the way up to the DOM root.
- d. Build Verification Opens the verification builder UI.
- e. Quick Tasks Quickly record verification or wait for a specific property on the element.



- f. Drag & Drop Enables you to record a drag & drop step.
- g. JavaScript Events Invoke a specific JavaScript event against the highlighted element.
- h. Mouse Actions has options to record specific desktop actions against the highlighted element.
- i. Scroll Element Adds a scroll element to the top or bottom of the page step.

Note: The Overlay Automation Surface allows users to record steps that they may not be able to record by simply interacting with the page e.g. mouse hover over.

Element Explorer Tool Window Overview



The "Elements Explorer" tool window maintains a list of all Elements within the current project. It provides a one stop shop to help maintain elements and the way they are found during execution. The Elements Explorer menu bar has these buttons:

- A green icon at the far left that controls highlighting of elements on the recording surface as they are selected in the tree view. Clicking on the button turns on element highlighting. Clicking on the button again turns off element highlighting.
- Refresh Clicking on this icon refreshes the display of the elements in Elements Explorer.
 You seldom should have to do this because Design Canvas normally automatically refreshes the window properly.
- c. View All or View Current Page only.

The tree view is organized by Page -> Frame -> Test Regions -> Elements. The hierarchy is maintained according to where the element was located on the page. For example, if there are no frames or regions then elements for that particular page will be listed under the "Page" node. Each Page Node has a context menu with two choices:

- a. Load Page loads the page in the recording surface.
- b. Validate All Elements validates that all elements can be located on the current page using the current FindParam settings.

Each Element node has a context menu with three active choices (when the page is loaded in the recording surface):

- a. Edit... allows you to change the Find Expression of the Element (the way the element is found).
- b. Locate In DOM Explorer highlights the element node in the DOM Explorer tool window.
- c. Load Page... loads the URL the element belongs to into the recording surface window.

Clicking each element will display the properties for that element in the Properties tool window.

Test Explorer Tool Window Overview

Test Explorer - WebAiiTest1										
🖉 🦊 🚹 🤊 🍽 Browser: Internet Explorer 🔹 🚀 💷 🚽 🍪 🦓 🛛										
	Order	Enabled	Description							
4	1	V	Navigate to : 'http://www.google.com/'		×					
4	2	V	Set 'QText' text to 'ArtOfTest'		×					
4	3	V	Click 'BtnGSubmit'		X					
P	4	V	Verify 'TextContent' 'Contains' 'ArtOfTest2' on 'ArtOfTestEmTag'		X					
	st Exp 2 4 7 7 7 7 7 7	Image: Constraint of the second sec	Image: Constraint of the second sec	Image: Set 'QText' text to 'ArtOfTest' Image: Set 'QText' text to 'ArtOfTest' Image: Set 'QText' text to 'ArtOfTest' Image: Set 'QText' text to 'ArtOfTest'	Image: Second					

The "Test Explorer" tool window provides the list of steps contained within the currently selected test. Each step has:

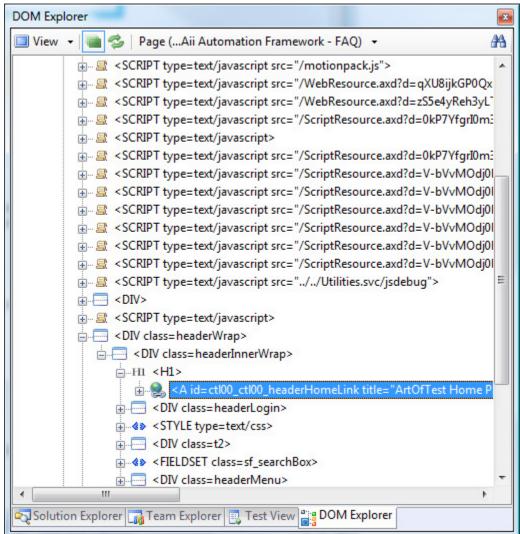
- a. Type Icon represents the type of step: Action, Verification, Coded.
- b. Order of the Step.
- c. Enabled checkbox whether the step will run during execution.
- d. Step description.
- e. Continue on failure indicator controls whether or not the test will stop if that verification step detects a failure.
- f. Delete Button deletes the step from the test.

The menu bar for the "Test Explorer" contains (from left to right):

- a. Clear all test steps deletes all steps from test.
- b. Move Step Down moves selected step down one in the sequence.
- c. Move Step Up moves selected step up one in the sequence.
- d. Undo Restores a step you just deleted or changed to its previous state & location.
- e. Redo Re-applies the edit that was undone with the Undo button.
- f. Browser Drop Down selects which browser to use for quick execution.
- g. Enable Test Annotation enables test annotation during Quick Execution. If you want to change the execution across all your tests then you need to change the setting in the test configuration as previously explained.
- h. Execution Delay sets (in milliseconds) the time to delay between each step execution.
- i. Quick Execute immediately executes the current test in the selected browser.
- j. Clear execution results clears the previous execution results of the last Quick Execute run.
- k. Add... This dropdown allows you to add special steps that can't be recorded. These steps include:
 - a. Capture Desktop Captures a screenshot of the desktop and stores it in a file.
 - b. Capture Browser Captures a screenshot of the browser window and stores it in a file.
 - c. Custom Annotation Allows you to add your own annotations to the test.

- d. Test as Step With this option you can include another test as a test step in this test. Design Canvas will run the other test when it comes to this step and, when complete, resume running the rest of the steps of this test.
- e. Wait x msecs Causes the test to pause for number of milliseconds you specify.
- f. Clear Cookies Will clear all of the browsers cookies.
- g. Wait for Url Waits for a specific URL to appear in the navigation bar of the browser. This is very useful for page redirection or form posting where there can be delay in processing by the web server.
- Inspection Point Pauses the test and opens a DOM window with the current DOM. You can then study the DOM tree. The test will automatically resume when you close this DOM window.

Clicking a step will display the properties for that step in the Properties tool window. You can edit the properties of that step (e.g. the URL to navigate to, the text to enter into a text box, check or uncheck a checkbox, etc.).



DOM Explorer Tool Window Overview

The "DOM Explorer" tool window displays the Document Object Model for the currently loaded page in the "Recorder" document window. The nodes are a hierarchical representation of all the HTML/XAML elements that make up the page. Each node in the tree is listed by:

<[Tag Name] [attributes]>

Clicking on a node in the DOM Explorer will highlight that element (if visible) in the Recorder document window. Right clicking a node will bring up a context menu containing two options:

- a) Goto provides a shortcut for moving to a significant layer of the DOM tree (Page root element, embedded frames, Silverlight applications).
- b) Add to Project Elements will add the currently selected element to the "Elements Explorer" tool window and craft a default FindParam for that element.

The menu bar for the DOM Explorer contains (from left to right)

- a) View (two options)
 - a. Hierarchal View Displays the DOM in the order of the page.
 - b. TagName Displays the DOM ordered by tag names.
- b) Refresh DOM will rebuild the tree from the current page.
- c) Move to main element drop down Lists the root element, any frames and any Silverlight applications found on the current page. Selecting one of these from the drop down highlights that element in the DOM Explorer tool window.
- d) Search will enable another menu below to search through the DOM tree.
- e) Find expression text box enter a valid Find Expression
 (<u>http://www.artoftest.com/support/webaii/topicsindex.aspx?topic=byexpression</u>).
- f) Evaluate Expression The Go button
- g) Move to Previous Result
- h) Move to Next Result

WebAii Test Tab Overview



Each WebAii test loaded in Visual Studio will have its own document tab allowing for step visualization and test data manipulation. The test tab has a menu bar with four icons:

- a) Add a code behind file Adds a new .cs or .vb code behind file which is then logically attached to the test in Solution Explorer.
- b) Convert test to VS WebTest Generates a Visual Studio WebTest from the test and adds it to the project.
- c) Generate Unit Test from WebAii Test Generates a Visual Studio Unit Test from the test.
- d) Record Opens the "Recorder" window and directly enables recording for this test)

Note: Some recorded steps will not have an image associated with them because of the nature of the action. For example, a "Navigate To" action does not have an image

Clicking on any of the step images in the "Storyboard" will bring that image forward and move to that step in Test Explorer. Each recorded image will have the element that it was recorded against highlighted within the image.

The test tab also contains a "Data" tab. The Data tab allows for the creation of a simple Excel like data array to be used by the steps of the test. See "Creating a Data Driven Test" section in the Design Canvas User's Guide for more details.

Customizing Your Test Using C# or VB.NET Code

Now that you have the core of your test put together, you may run into a scenario that requires a test step that is more complex than can be put together using the Verification Builder or the actions provides by the Element Toolbox. Automation Design Canvas supports using a "code behind file". Using a code behind file you can actually write any code you need and have that code executed as a test step. The following sections describe how to do that.

📸 TestProject1 - Microsoft Visual Studio (Administrator) _ 0 X <u>File Edit View Refactor Project Build Team Debug Data Tools Test Analyze Window Help</u> |罰・罰・醫・胃||誤 🍓 🖫 🚳 🐙 🕄 | 🆘 🆘 | 図 🖉 🗳 🖬 🖬 🖬 🖬 🖉 🤹 👘 🖏 😓 🐼 | 筆 第 | 三 😫 | ロ 🖓 🤜 🗛 🗛 🦓 Test View ЛХ - × WebAiiTest1.aii.cs WebAiiTest1.aii Recorder (WebAiiTest1) Start Page Source Control Explorer Test List Editor CestProject1.WebAiiTest1 -(i) Item(s) selected: 1 ActiveBrowser.NavigateTo("<u>http://www.google.com</u>"); 34 35 11 Test Name Project 36 11 // Or ManualTest1 TestProject1 37 11 ActiveBrowser.NavigateTo(Data["url"]); 📄 TestMethod1 TestProject1 11 } 🔒 WebAiiTest1 TestProject1 39 40 11 41 42 public class WebAiiTest1 : BaseWebAiiTest 43 44 + [Dynamic Pages Reference] 66 • 111 67 [CodedStep(@"Set 'QText' text to 'WebAii'")] 68 public void WebAiiTest1 CodedStep() Test Vi.. 🟹 Solutio... 🚮 Team ... 69 DOM Explorer - 4 × 70 🔲 View 👻 🐚 🤧 71 // Set 'QText' text to 'WebAii' 72 Pages.Google.QText.Text = "WebAii"; 73 74 3 75 76 77 78 // Add your test methods here ... 3 - } 79 Properties ЛХ Test Explorer - WebAiiTest1 - 4 × Elements Explorer - I X 📿 👢 👔 🤊 🕲 Browser: Internet Explorer - 😳 🗐 🐚 🤣 📴 View (All) - 🧭 400 >> 8 **2** | C 🖃 🗐 🗐 All Pages Order Enabled Description -- 🝙 Google 4 1 1 Navigate to : 'http://www.google.com/' × BtnGSubmit 43 1 × 2 [WebAiiTest1_CodedStep] : @"Set 'QText' text to 'WebAii' QText 4 Click 'BtnGSubmit' × 3 🎝 Breakpoints 📰 Test Results 🖏 Find Results 1 🙀 Find Symbol Results 🧐 History 蹪 Error List 🚯 Pending Changes 🔳 Output Ready Col 2 Ch 2 INS Ln 78

Creating a Test With a Code Behind File

- 1) Create a test as outlined above.
- 2) Record a few steps as outlined above.
- 3) There are two ways of creating a code behind file for your test.
 - a. On the test tab, click the "Add code behind..." button.



- b. Right click on any step in the "Test Explorer" and select "Convert To Code".
- 4) Method A will create a blank code behind file while method B will create a code behind file with the selected step converted into a method matching the selected step name.
- 5) Note that a step with generated code will be read only in the "Test Explorer" tool window. The description can only be changed in the "CodedStep" method attribute that appears in the code behind file. A coded step will be indicated with a code icon and will have a white background in the "Test Explorer" tool window.

Test Explorer - WebAiiTest1 🗸 🗸 🧎										
4	2 1	1	9 (~	Browser: Internet Explorer	- 🥩 400	-		>>		
	Order Enabled Description									
	4	1	V	Navigate to : 'http://www.g	Navigate to : 'http://www.google.com/'					
	A	2	V	[WebAiiTest1_CodedStep] :	WebAiiTest1_CodedStep] : @"Set 'QText' text to 'WebAii'					
	4	3	V	Click 'BtnGSubmit'				X		

- 6) Once a step has been converted to code, you cannot convert it back to a regular step.
- 7) Save and build the project.
- 8) Execute the test.

Creating a Test With Custom Code Steps

° TestProject1 - Microsoft Visual Studio (Administrator) <u>File Edit View Refactor Project Build Team Debug Data Tools Test Analyze Window Help</u> 🛅 • 🛅 • 📂 🔒 Test View ЛХ WebAiiTest1.aii.cs WebAiiTest1.aii Recorder (WebAiiTest1) Start Page Source Control Explorer Test List Editor - × TestProject1.WebAiiTest1 MvCustomStep() -(i) Item(s) selected: 1 ActiveBrowser.NavigateTo("<u>http://www.google.com</u>"); 34 35 Test Name Project 36 11 // Or 🔬 ManualTest1 TestProje 37 11 ActiveBrowser.NavigateTo(Data["url"]); 🛃 TestMethod1 TestProie 38 11 } 🔒 WebAiiTest1 TestProje 39 40 41 42 public class WebAiiTest1 : BaseWebAiiTest 43 [Dynamic Pages Reference] 44 66 // Add your test methods here ... 61 📑 Test... 🟹 Solu... 🏹 Tea... 68 69 [CodedStep(@"My Custom Coded Step")] DOM Explorer 🚽 🕂 🗙 70 public void MyCustomStep() 🔲 View 👻 🐚 🤣 🛛 71 72 73 74 75 Properties - I X Test Explorer - WebAiiTest1 - I X Elements Explorer - 1 X 📿 🖡 👔 🤊 🛯 Browser: Internet Explorer - 🍪 📄 Add... -🐚 🤣 📴 View (All) - 🥩 400 --- 🗐 All Pages 8 2↓ 🖻 Order Enabled Description 🗄 🛛 💽 🔂 🖬 1 Navigate to : 'http://www.google.com/' × 1 BtnGSubmit 4 2 × 1 Set 'QText' text to 'WebAii' QText 4 V Click 'BtnGSubmit' × 3 43 4 [MyCustomStep] : @"My Custom Coded Step × 🎝 Breakpoints 📴 Test Results 🖼 Find Results 1 🙀 Find Symbol Results 🧐 History 👶 Error List 📑 Pending Changes 📃 Output Item(s) Saved Col 1 Ch1 INS

- 1) Create a test and add a code behind file using method A as outlined above.
- 2) Just like a step that has been converted to code; a method in the code behind file is represented as a step in the "Test Explorer" tool window.
- 3) In order to write custom steps in code, you must add a method in the code behind file that takes no parameters and has a "void" return type.
- 4) The method must be decorated with the "CodedStep" attribute as shown below:

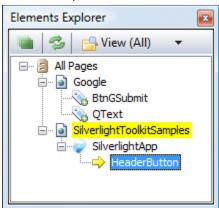
```
[CodedStep("MyCustom Step Description")]
public void MyCustomStep()
{
    // Custom code goes here
    ActiveBrowser.NavigateTo("http://www.google.com");
}
```

- 5) Add the above method or one similar to your liking to your code behind and click save.
- 6) Notice that the coded step is added to the "Test Explorer" with the coded icon immediately upon saving. NOTE: Newly coded steps are always added to the end of the steps in the "Test Explorer" tool window. You can use the move up/down buttons on the "Test Explorer" tool window to change its order of execution or drag and drop the step to the desired position.

- 7) The name of the method becomes the step name and the description from the Coded Step attribute is listed as the description in the "Test Explorer" tool window.
- 8) Save and build the project.
- 9) Execute the test.
- 10) The code behind file has access to all the WebAii runtime objects like ActiveBrowser... etc. If you are familiar with the WebAii runtime automation framework, this will be an identical coding experience.

How to Reference Elements from the Element Explorer in Code Behind Files

1) Pages can be referenced in the code behind file in the same way they are found hierarchically in Elements Explorer.



- 2) For example: If the image above was your current Elements Explorer window, you could reference the "BtnGSubmit" element like this:
 - a. Pages.Google.BtnGSubmit
- 3) In this example it will return the strongly typed WebAii framework "HtmlInputSubmit" type.