telerik
deliver more than expected

Understanding how Test Studio differs from Visual Studio 2010

This page is designed to help clarify the way Telerik Test Studio goes beyond Visual Studio 2010 testing products (Visual Studio Test Premium 2010 and Visual Studio 2010 Ultimate) to offer richer user experience and enhanced productivity. Its purpose is also to help visitors understand the key challenges with test automation that we are trying to solve with our solution beyond just the "Hello World" test case.

With the release of Visual Studio 2010, Microsoft has introduced the capability to automate the user interface (UI) of applications on the Windows platform. Visual Studio 2010 introduces the CodedUI test framework and the MTM (Microsoft Test Manager) as solutions for Testers and QAs to help manage test creation and test automation. We believe this to be a great step forward and a win for our customers especially in the areas of Test and Lab management.
The Visual Studio / .NET platform has traditionally been exclusive to developers but now with VS2010 it also encompasses testers and QA professionals.

Quick Facts Sheet:

Benefits	**Test Studio**	**Visual Studio 2010**
Level of complexity	Technical & Non-Technical	Development skills required
Point-and-click UI	Yes	Limited
Easy to Maintain Tests	Yes	No
Silverlight automation	Yes	Yes, only for Silverlight 4
Test step keyword view	Yes	No
Browser support	IE, FF, Safari, Chrome	IE, FF
Native support for Telerik controls for AJAX and Silverlight	Yes	No
Web-specific testing features	Yes	Yes
Licensing model	Integrates with VS Professional and above	Visual Studio Ultimate / Premium Edition license
Major product updates	3 times a year	Unknown
Premier Support (response within 24 hours)	Yes	No

Contents

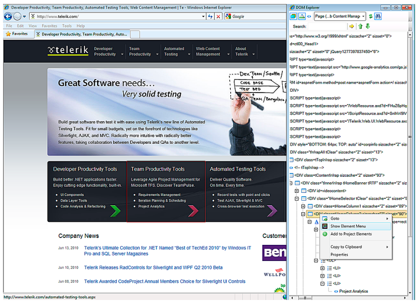
Are you looking for a specialized web testing platform that will allow your regression suite to grow with time?

The Visual Studio 2010 approach to test automation is to offer a broad solution that is a good entry point for developers looking to begin test automation. As your automation needs grow, you may run into challenges that arise from Coded UI's single approach for covering multiple technologies.

With Test Studio we address the specific automation needs of web applications. We studied the workflow of testers when automating for the web and the entire testing life-cycle which helped us identify areas that take the longest to automate or are very tedious. We've built many specialized features that cater to these needs that you will not find in VS 2010, here are a few worth noting:

DOM Explorer

A full DOM explorer for your web application that supports nested frames and allows you to do rich search across the entire DOM. You can also navigate directly to specific nested Frame or iFrame with one right click and directly craft verification from it. This feature reduces your reliance on external tools and reduces the context switching between our tool and others.



Developer Productivity, Team Productivity, Automated Testing Tools, Web Content Management | Te - Windows Internet Explorer

http://www.telerik.com/

Great Software needs… *Very solid testing*

Build great software then test it with ease using Telerik's new line of Automated Testing Tools. Fit for small budgets, yet on the forefront of technologies like Silverlight, AJAX, and MVC. Radically more intuitive with radically better features, taking collaboration between Developers and QAs to another level.

Developer Productivity Tools

Build better .NET applications faster. Enjoy cutting edge functionality, built-in.

- UI Components
- Data Layer Tools
- Code Analysis & Refactoring

Team Productivity Tools

Leverage Agile Project Management for Microsoft TFS. Discover TeamPulse.

- Requirements Management
- Iteration Planning & Scheduling
- Project Analytics

Automated Testing Tools

Deliver Quality Software. On time. Every time.

- Record tests with point and clicks
- Test AJAX, Silverlight & MVC
- Cross-browser test execution

Company News

Jun 13, 2010	Telerik's Ultimate Collection for .NET Named "Best of TechEd 2010" by Windows IT Pro and SQL Server Magazines
Jun 10, 2010	Telerik Releases RadControls for Silverlight and WPF Q2 2010 Beta
Jun 08, 2010	Telerik Awarded CodeProject Annual Members Choice for Silverlight UI Controls

http://www.telerik.com/automated-testing-tools.aspx

```
s="http://www.w3.org/1999/xhtml" sizcache="2" sizset="0">
d=ctl00_Head1>
sizcache="2" sizset="0" jQuery1277397837450="6">
RIPT type=text/javascript>
RIPT type=text/javascript src="http://www.google-analytics.com/ga.js
RIPT type=text/javascript>
RM id=aspnetForm method=post name=aspnetForm action=/ sizcach
DIV>
SCRIPT type=text/javascript>
SCRIPT type=text/javascript src="/WebResource.axd?d=FHuZ6pHq
SCRIPT type=text/javascript src="/ScriptResource.axd?d=SnlhW9lV
SCRIPT type=text/javascript src="/Telerik.Web.UI.WebResource.ax
SCRIPT type=text/javascript>
SCRIPT type=text/javascript>
DIV style="BOTTOM: 64px; TOP: auto" id=corpinfo sizcache="2" siz
DIV class="tWrapAll tClear" sizcache="2" sizset="13">
<DIV class=tTopWrap sizcache="2" sizset="13">
<!-- tTopWrap -->
<DIV class=tContentWrap sizcache="2" sizset="93">
 <DIV class="tInnerWrap tHomeBanner tRTF" sizcache="2" siz
   <DIV id=videocontent>
   <DIV class="tHomeSelector tClear" sizcache="2" sizset="
     <DIV class=tHomeColumn1 sizcache="2" sizset="89">
     <DIV class=tHomeColumn2 sizcache="2" sizset="90">
                              pe=rect
         <UL>
           <LI>
           <LI>
           <LI>
             Project Analytics
```

Enlarge image

Web Verifications

Build computed or in-line style verifications directly from the UI in addition to the attribute/content and CSS visibility verifications. Such scenarios are important when validating web element positioning and layering.

DOM Captures on Failure

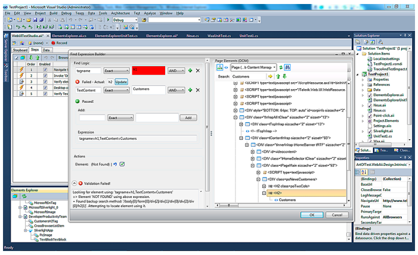
When your test fails, we capture the DOM of the page at the state of failure and display it for you to help you resolve your failure easier.

Execution Intelligence

The execution engine is built to handle AJAX solutions from the ground-up and if you are executing a web test, the engine will intrinsically ensure the element exists and is visible on the page before running each step. The check for visibility uses the actual HTML/CSS visibility checks rather than UI Automation, which is much more reliable. For Silverlight scenarios we also ensure the element is not moving by doing a motion check on the actual visual element. This feature enhances the reliability of your execution dramatically so that you don't need to clutter your tests with sleeps and waits.

Find Expressions & Live Validation

As many experienced testers will tell you, tests frequently fail due to element changes in the web application as it



TestProject1 - Microsoft Visual Studio (Administrator)
File Edit View Project Build Debug Team Data Tools Architecture Test Analyze Window Help
Debug
WebUITestStudio.aii* ElementsExplorer.aii.cs ElementsExplorerUnitTest.cs ElementsExplorer.aii* Neue.cs WaaUnitTest.cs UnitTest1.cs
(none) Record
Storyboard Steps Data
Order Enabled
Server Explorer Toolbox
Find Expression Builder
Find Logic
Page Elements (DOM)
Page (...b Content Manag
Add
OK Cancel
Elements Explorer
Solution Explorer
Properties
Ready

Order	Enabled	
1	☑	Navigate t
2	☑	Invoke 'Or
3	☑	Verify elen
4	☑	Desktop co
5	☑	Verify Tex

tagname Exact h1 AND

Failed - Actual: h2 Update

TextContent Exact Customers AND

Passed!

Expression: tagname=h1,TextContent=Customers

Actions: Element: (Not Found)

Validation Failed!

Looking for element using: 'tagname=h1,TextContent=Customers'
=> Element 'NOT FOUND' using above expression.
+ Found backup search method: '/body[0]/form[0]/div[2]/div[1]/div[0]/div[2]/div[0]/h2[1]'. Attempting to locate element using it.

Search: Customers

```
<SCRIPT type=text/javascript src="/ScriptResource.axd?d=SnlhW
<SCRIPT type=text/javascript src="/Telerik.Web.UI.WebResource.
<SCRIPT type=text/javascript>
<SCRIPT type=text/javascript>
<DIV style="BOTTOM: 64px; TOP: auto" id=corpinfo sizcache="2"
<DIV class="tWrapAll tClear" sizcache="2" sizset="13">
  <DIV class=tTopWrap sizcache="2" sizset="13">
  <!-- tTopWrap -->
  <DIV class=tContentWrap sizcache="2" sizset="93">
    <DIV class="tInnerWrap tHomeBanner tRTF" sizcache="2"
      <DIV id=videocontent>
      <DIV class="tHomeSelector tClear" sizcache="2" sizset
      <DIV class=tPageMain sizcache="2" sizset="93">
        <SCRIPT type=text/javascript>
        <DIV class=poNewsCustomers>
          <H2 class=poTwoCols>
          <H2>
            Customers
```

Elements Explorer: MicrosoftEmTag, MicrosoftSilverlight_0, MicrosoftImage, DeveloperProductivityTeam, CustomersH2Tag, CrossBrowserListItem, SilverlightApp, PicImage, TextBlockTextblock

Solution Explorer: Solution 'TestProject1' (1 proj, Solution Items, Local.testsettings, TestProject1.vsmdi, TraceAndTestImpact.t, TestProject1, Properties, References, Data, ElementsExplorer.aii, ElementsExplorerUnitT, Neue.aii, Neue.cs, Point-click.aii, Project.Elements, Settings.aiis, Silverlight.aii, UnitTest1.cs, Waa.aii

Properties: ArtOfTest.WebAii.Design.Intrinsic

(Bindings)	(Collection)
BaseUrl	
ClosesBrowse	False
LogMessageC	
NavigateUrl	http://www.tel
Pause	None
PrimaryTarge	
RunsAgainst	AllBrowsers
SecondaryTar	

(Bindings) Bind data driven properties against a datasource. Click the drop down t...

evolves. The Find Expression is a UI editor that allows you to easily resolve element find failures with a suggest feature to help you understand why the find failed. It also comes with an integrated DOM so that you can validate right there and then that you are finding the correct element instead of having to execute the test with each edit. A great time saver.

Enlarge image

Coded vs. Code-less (script-less) Automation

Do I always have to write code to automate my test? What if my tester is not proficient in C# or VB.NET? Am I stuck with manual testing?

We don't believe you always need to write code to build test automation. In Visual Studio 2010, to properly perform test automation of real line-of-business applications you always have to revert to the CodedUI test which is a pure C# or VB.NET class that leverages the CodedUI framework for automation. The element mapping in Visual Studio 2010 - also known as the UIMap - generates a complex hierarchy of classes to represent the elements as strongly-typed objects in code. This approach is not very convenient when it comes to test maintenance. For example, a simple 15-step test in VS2010 generates around 1000 lines of code for the test method.

Yes, there will always be cases when you will need to revert to code but that doesn't mean your entire test has to be created and / or maintained in code.

Test Studio has been designed to allow most QA/Test professionals to automate without having to revert to code in most scenarios. Even complex conditional logic, element extraction, and data driven tests can be created and maintained without writing a single line of code.

When complex logic is needed, a step or two out of your entire test can be converted and written in C# or VB.NET code without having to convert the entire test. Such approach not only simplifies test automation for non-developer QA professionals but also makes test maintenance a lot easier. You can update your test by changing properties rather than editing code and doing code refactoring. Many of our customers that use Test Studio end up writing 90-95% of their test automation without having to code anything. Minimizing complex code will increase your QA Teams productivity, allowing more time to create a larger regression suite resulting in better software, delivered on time and on budget.

UIAutomation vs. Full Application Access

UIAutomation technology is the evolution of the Accessibility API which is the mechanism by which screen readers and other aided technologies enable visually-impaired users to interact with a computer. Since 2005 Microsoft have been pushing UIAutomation as a solution not only for accessibility tools but also for test automation.

UIAutomation depends on application level hooks that give access to external programs to invoke or retrieve information from the application. UIAutomation is 100% dependent on the developer of the application giving access to these parts of the application; else those parts can't be automated. With UIAutomation we found ourselves constantly hitting the wall of "not exposed by developer" argument and we ended up having to run tests manually because developers didn't have enough cycles to expose needed functionality.

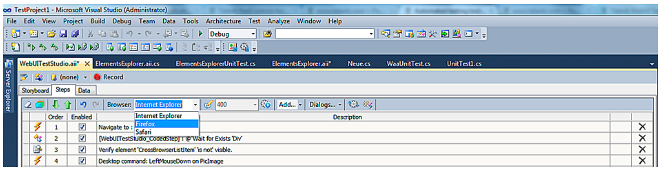
Furthermore, UIAutomation does not offer real-user simulation. When calling a UIAutomation API you are actually invoking a method call within your application. The user experience could actually be a lot different for many applications.

When building Test Studio, we wanted to build a solution for testers that gives them freedom and full access to the application under test without limitations. Therefore we completely avoided any heavy reliance on UIAutomation and used it only in certain scenarios. Our architecture relies on having full access to the application whether it is a Silverlight application or an HTML page. You can access any and every element in the application and perform real-user actions against it.

The Visual Studio 2010 automation solution is almost completely architected on UIAutomation.

Cross-Browser Support

With the release of VS2010, the matrix support for browsers includes IE & Firefox. Test Studio currently supports IE, Firefox, Safari, and Google Chrome.


TestProject1 - Microsoft Visual Studio (Administrator)
File Edit View Project Build Debug Team Data Tools Architecture Test Analyze Window Help
Debug
WebUITestStudio.aii* ElementsExplorer.aii.cs ElementsExplorerUnitTest.cs ElementsExplorer.aii* Neue.cs WaaUnitTest.cs UnitTest1.cs
Server Explorer
(none) Record
Storyboard Steps Data
Browser: Internet Explorer
Internet Explorer
Firefox
Safari
400 Add... Dialogs...

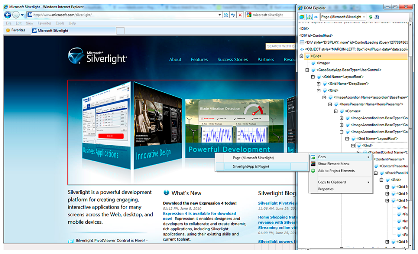
Order	Enabled	Description
1	☑	Navigate to :
2	☑	[WebUITestStudio_CodedStep] : @'Wait for Exists 'Div'
3	☑	Verify element 'CrossBrowserListItem' 'is not' visible.
4	☑	Desktop command: LeftMouseDown on PicImage

Enlarge image

Silverlight Automation

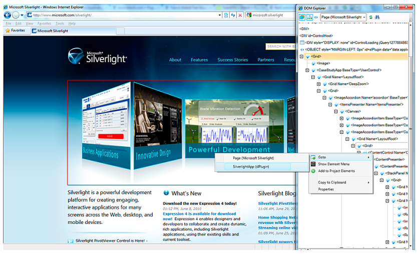
VS2010 Coded UI has just released a feature pack that includes support of Silverlight 4 applications. Test Studio has been automating against Silverlight since version 2 and has made many refinements in our record/playback compatibility with rich specialized features for Silverlight applications automation including:

- **Visual Tree inspection:** View the entire visual tree of your Silverlight application and target specific elements for verifications.



Enlarge image

- **Rich support for out-of-box controls**: All the out-of-box controls that ship with Silverlight are supported by Test Studio with specialized translators for each control. For example, when using the Silverlight Calendar control and selecting a date, our translators understand the control being used and can determine the action you are performing. So instead of recording a click against a low-level element, it records a select for a specific date. This makes the test easy to understand, and easy to update.



Enlarge image

- **Custom controls support:** From experience with real-business applications that customers are trying to automate, we found that many applications use custom controls built in house by the development team. Some inherit from Panels/Calendars/Checkboxes…etc. Our tool provides a smart control matching feature that allows the tool to drop to the base control type and offer the common tasks for that control instead of dropping to just a raw base element with only generic tasks associated with it.

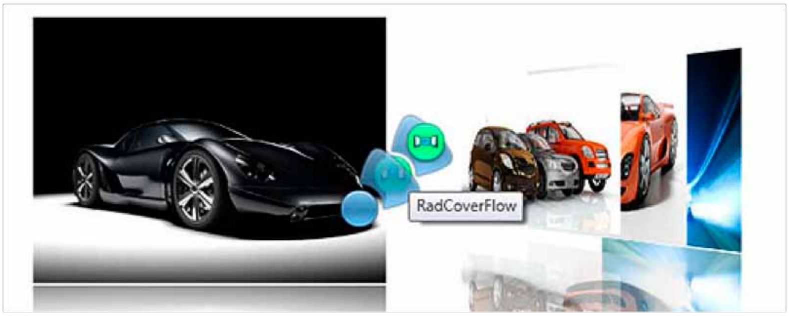
Read more about Silverlight testing with Test Studio

Native Support for RadControls for AJAX and Silverlight

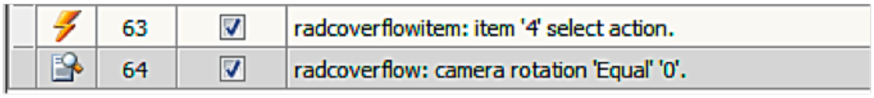
If you use RadControls (*ASP.NET AJAX* or *Silverlight*) in the application you are trying to automate, then you won't find a more suited test automation tool for your application than Test Studio.

Test Studio comes with rich support for each RadControl:

- RadControls are automatically detected on your page visually.

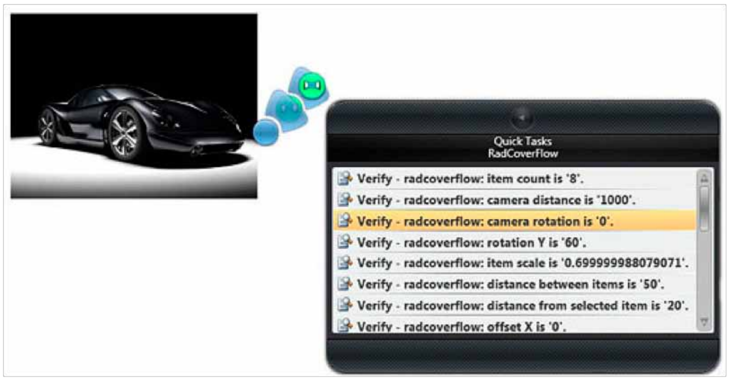


- When performing an action within the control, the action is routed through a special translator built for each control to help identify the action at higher contextual level than a raw click, for example: TreeView=>Expand, Calendar=>SelectDate, DropDown=>Select Item



⚡	63	☑	radcoverflowitem: item '4' select action.
	64	☑	radcoverflow: camera rotation 'Equal' '0'.

- Each control has a set of specialized verification and synchronization tasks that are built dynamically for your application in the desired state you want verified:



Verify - radcoverflow: item count is '8'.
Verify - radcoverflow: camera distance is '1000'.
Verify - radcoverflow: camera rotation is '0'.
Verify - radcoverflow: rotation Y is '60'.
Verify - radcoverflow: item scale is '0.699999988079071'.
Verify - radcoverflow: distance between items is '50'.
Verify - radcoverflow: distance from selected item is '20'.
Verify - radcoverflow: offset X is '0'.

With Test Studio you will have the ability to:

- Spend more time building test scenarios to ensure the quality of your application rather than trying to figure out how to automate an action on the control and understanding the complexity of its rendering.
- You don't need to worry about backward compatibility. With each RadControls release, the translators are updated so your test continues to work without having to touch any of your tests.

Customer Support & Release Agility

At Telerik we have *world-class* support that starts the day you download your trial. Each trial user gets the ability to not only use the community forums to ask questions but also gets to submit support tickets that our team of support engineers address promptly, with friendly service and knowledgeable responses.

Each license comes with 1 year of support and updates. Each ticket submitted is responded to within 24 hours. If you are still having problems, we schedule GoToMeetings to help guide you through you automation scenario.

With three major releases a year and weekly public internal build releases, your feature requests and bug fixes are accessible with a faster turnaround unmatched by any other product.