

Microsoft® Virtual Labs

Developer Tools in Internet Explorer 8:
Debugging and Application Compatibility

Table of Contents

Developer Tools in Internet Explorer 8: Debugging and Application Compatibility	1
Exercise 1 CSS Debugging with IE8 Developer Tools	2
Exercise 2 JavaScript Debugging with IE8 Developer Tools	5
Exercise 3 Web Application Compatibility in IE8	9

Developer Tools in Internet Explorer 8: Debugging and Application Compatibility

Objectives

After completing this lab, you will be better able to:

- Test live web pages for compatibility in IE8
- Use IE8's developer tools to debug CSS and JavaScript issues

Scenario

As a developer for Northwind Traders, you've been tasked with resolving display and scripting issues on the company's website. The website makes extensive use of CSS for content display and styling but recent changes to the site's style sheet have caused unexpected results on certain pages. There is also a minor issue with a JavaScript image viewer.

Northwind Traders' IT staff has deployed Internet Explorer 8 to all its developers for testing the company's web site and online applications to ensure full compatibility. You will use IE8's developer tools to find the CSS errors and correct them.

Each product page contains multiple images of the product in thumbnail view and the option to view them as a larger image. This is all driven by JavaScript that is used site wide but, after recent updates, the script has stopped functioning. You will use IE8's built-in JavaScript debugger to resolve this issue with the image viewer.

Finally, you've been tasked with testing the Northwind Traders pages in IE8 to ensure that any compatibility problems are resolved before IE8 is available to the public as a stable release. You will use IE8's browser and document mode features to test for any compatibility issues.

To save time, certain components have been pre-installed and pre-configured. These components include Internet Information Services 7 (IIS7) and Internet Explorer 8. You will be using IE8's developer tools for HTML, CSS, and JavaScript debugging.

For the purposes of this lab, you will be developing locally and using the local copy of IIS for testing.

Estimated Time to Complete This Lab

45 Minutes

Computers used in this Lab





The password for the Administrator account on all computers in this lab is: P@ssword.

Exercise 1


CSS Debugging with IE8 Developer Tools

Scenario

In this exercise, you will use IE8's developer tools to resolve CSS issues on a website.

Tasks	Detailed Steps
<p>Complete the following tasks on:</p>  <p>1. Viewing the Problem</p>	<p>a. Start off by opening Debugging.html in IE8. If the ActiveX security warning bar appears allow the blocked content. This is a product page of Northwind Traders that is broken due to CSS issues. As shown in Figure below, the navigation menu displays correctly but the content section with the item description and images does not. It appears below the navigation menu instead of directly to the right.</p> 
<p>2. Using Developer Tools to Find the Problem</p>	<p>a. Opening Developer Tools is very easy. Just press F12 and it will open in a new window. You may also access Developer Tools by going to Tools > Developer Tools.</p> <p>Note: If Developer Tools opens in a new window and you would prefer to view the Developer Tools pane within the browser window, click the "Pin" button on the upper right hand corner of the window. (see Figure below)</p>

Tasks	Detailed Steps
	<div data-bbox="609 191 1328 390" data-label="Image"> </div> <p data-bbox="505 401 1430 653">b. Once Developer Tools is open click Find > Select element by click. This will enable you to find code blocks based on their visual element. The error we are encountering involves the navigation menu as well as the content block so we will start looking at the code behind the navigation menu. Hover over the dark brown line above the navigation menu. You will see a blue border appear around the entire navigational menu. While hovering over the dark brown area, click once and the corresponding section in the source will be highlighted within the Developer tools pane. (see Figure below)</p> <div data-bbox="509 659 1419 1268" data-label="Image"> </div> <p data-bbox="505 1283 1430 1377">c. With the div block “sidebar” (which makes up the navigational menu) selected, the corresponding CSS that styles the block is now visible on the right hand side of the Developer Tools pane.</p> <p data-bbox="505 1388 1430 1482">d. Find the class “.sidebar” within the CSS and look at the properties associated with it. It has a font property, a brown border on the top (which was clicked earlier to select the navigational menu block) and a float property.</p> <p data-bbox="505 1493 1430 1619">e. The menu is floating left. A quick look at the content block will reveal that it is also floating left. Due to an undefined width, the navigational menu is taking up the whole width of the page and the content block is moved to the next line. The easiest way to fix this problem is to define a width for the navigational menu.</p>
<p data-bbox="186 1633 479 1696">3. Fixing the issue in the CSS file</p>	<p data-bbox="505 1633 1430 1759">a. Open the CSS file for this lab (Debugging.css) which is located in the C:\inetpub\wwwroot\DTLab\styles folder with Notepad running as Administrator. Look for the navigational menu class (.sidebar) and insert the following property below the line “float: left;” (see Figure below)</p> <div data-bbox="505 1766 1398 1791" data-label="Code-Block"> <pre>width: 141px;</pre> </div>


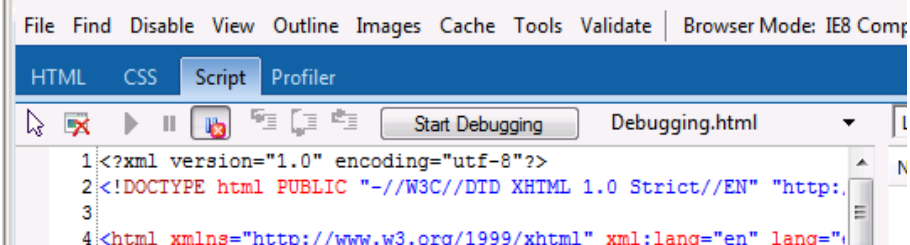
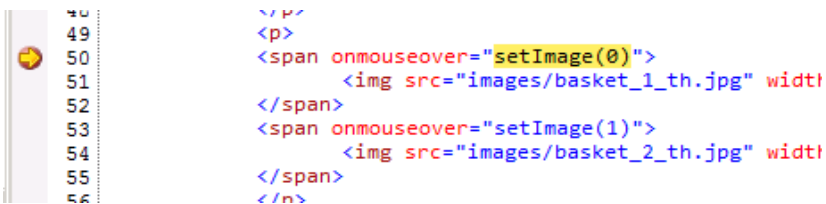
Tasks	Detailed Steps
	<pre data-bbox="511 210 1339 388"> .sidebar { float: left; width: 141px; border-top: 11px solid #8c4b00; font: 12px Arial, Helvetica, sans-serif; } </pre>
<p data-bbox="186 420 479 451">4. Reviewing the results</p>	<p data-bbox="511 420 1388 514">a. Save the file as debugging.css (all files not text) and go back to your browser window and click refresh. The content block should now be at the right of the navigational menu. (see Figure below)</p> <div data-bbox="511 556 1412 1050">  </div>

Exercise 2

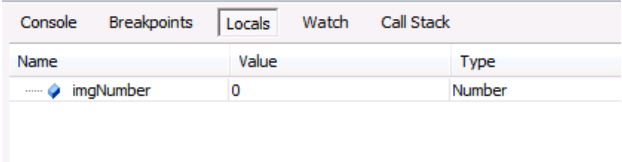
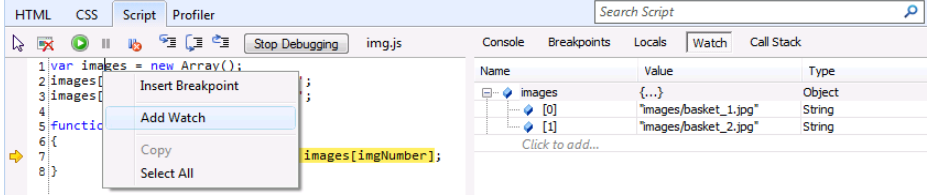
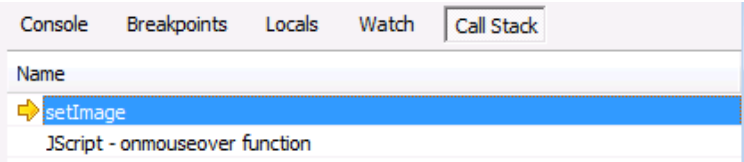
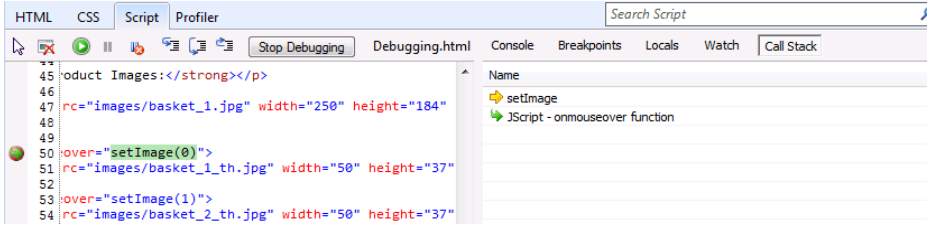
JavaScript Debugging with IE8 Developer Tools


Scenario

In this exercise, you will use IE8's developer tools to debug a JavaScript application.

Tasks	Detailed Steps
<p>Complete the following tasks on:</p>  <p>1. Using Developer Tools to Find the Problem</p>	<p>a. With Debugging.html open in IE8, click refresh. The image of the entire gift basket should be displaying.</p> <p><i>Note: If the first image is not displaying refresh the page again and keep your mouse away from the images block.</i></p> <p>b. You should be able to move your mouse over the small thumbnails to change the larger image. However, the image remains unchanged regardless of mouse movement. This is the general area where the problem is occurring.</p> <p>c. Open Developer Tools by pressing F12 or going to Tools > Developer Tools.</p> <p>d. Click on the Script tab to view the JavaScript debugger. Click the Start Debugging to start the IE8 debugger. A dialog box will appear, stating the webpage needs to be refreshed to begin debugging. Click OK. Note that the debugger will run in a separate window if you have the Developer Tools pinned inside the IE8 window.</p>  <p>e. Since the mouse over is not working correctly, let's begin there. Scroll down the HTML to find the onmouseover event inside a SPAN tag on line 50. Add a breakpoint by right-clicking any text on the line 47 and choosing Insert Breakpoint. Alternatively, you can click in the gray area directly to the left of the line numbers. (see Figure below, this is the location with the red dot).</p> <p>f. Now return to IE8 and move your mouse over the thumbnail on the left. No page refresh is required. As soon as you mouse over the image, the Developer Tools window regains focus, highlighting where in the page execution has been halted. Note that the debugger has stopped on line 50. It has not yet executed any code on line 50.</p>  <p>g. Here you can use the Developer Tool's features to control the execution of the code. Look at the debugger controls near the top of the window:</p>

Tasks	Detailed Steps
	<div data-bbox="764 201 1175 296" style="text-align: center;"> </div> <p data-bbox="505 310 1425 1031"> h. The debugger controls are similar to ones found in other IDEs such as Microsoft Visual Studio. They are as follows: i. Continue (F5). Continues to run the script without pausing, until another breakpoint or script error is encountered. j. Break All (Ctrl + Shift + B). Pauses execution immediately before the next script statement executes. k. Break On Error (Ctrl + Shift + E). Pause execution at the point where the error has occurred. l. Step Into (F11). Executes the next line of script and pauses, even if the next line is inside a new method. m. Step Over (F10). Continues to the next line of script in the current method, and then pauses. Useful for stepping over method calls. n. Step Out (Shift + F11). Continues executing the script to the next line in the method that has called the current method. Useful for stepping out of loops and method calls. o. The execution is stopped right before the execution of the onMouseOver event. Click Step Into (or hit F11) to go to where the function setImage() is defined. Notice that the debugger has moved from Debugging.html to img.js. All script files associated with a site are loaded into a dropdown menu directly to the right of the “Start/Stop Debugging” button. You can jump between different files easily. </p> <div data-bbox="683 1037 1252 1297" style="text-align: center;"> </div> <p data-bbox="505 1346 1425 1440"> p. Your screen should look similar to the screen shot below. Notice the five buttons in the upper right portion of the screen, highlighted by the red rectangle in the image. </p> <div data-bbox="509 1451 1414 1871" style="text-align: center;"> </div>

Tasks	Detailed Steps
	<p>q. Click on Breakpoints. This pane shows all breakpoints associated with the current website. You can disable and enable breakpoints by clicking the checkbox next to them.</p> <p>r. Click on Locals. This pane displays all variables that are currently within the local scope. Since execution is paused within the setImage function, the only local variable available is the parameter of the function, imgNumber. Notice that you can right click on a variable and edit its value directly from the debugger.</p>  <p>s. If there are too many variables for one pane, you can selectively choose which ones to watch using the Watch pane. Click on the Watch pane now.</p>  <p>t. To add a variable to the Watch pane, simply right click the variable name and select Add Watch. Try this now. Alternatively, you can type the name of variables into the Watch pane directly. You can watch variables that do not fall directly in the local scope of the paused script.</p> <p>u. The last area of interest is the Call Stack. These are the “breadcrumbs” tracing back through the script, with the newest location on top.</p>  <p>v. The yellow arrow indicates where the script is paused. You can see that execution is currently paused in the function setImage. Double-click on “JScript – onmouseover function” to show what called setImage. The debugger will highlight the calling statement in green.</p>  <p>w. Make sure that the Break on Error button is enabled and click Step Into to advance to the next statement. The debugger continues execution and stops on all errors. Line 7 contains the error, so the debugger notifies you with a balloon.</p>


Tasks	Detailed Steps
	 <p>x. It appears the error is on line 7. The entity document.images.rollimg does not have the .src property.</p>
<p>2. Fixing the issue in the JavaScript file</p>	<p>a. Now that the error has been found, open scripts/img.js in Notepad for editing.</p> <p>b. Locate the line inside the function setImage and add .src immediately after rollimg. (see Figure below)</p> <pre> var images = new Array(); images[0] = 'images/basket_1.jpg'; images[1] = 'images/basket_2.jpg'; function setImage(imgNumber) { document.images.rollimg.src = images[imgNumber]; } </pre>
<p>3. Reviewing the results</p>	<p>a. Save and close the file, return to the browser window, and refresh the page. Now hover over both thumbnail images several times. They should both work now.</p>


Exercise 3

Web Application Compatibility in IE8

Scenario

In this exercise, you will use IE8’s Developer Tools to test a website for standards compliance and to ensure full compatibility with IE8.

Tasks	Detailed Steps																
<p>Complete the following tasks on:</p>  <p>1. Understanding Browser Modes</p>	<p>a. With the Browser Mode menu, you can choose how the browser will report the following three properties:</p> <table border="1" data-bbox="508 636 1414 1129"> <thead> <tr> <th>Property</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>User Agent string</td> <td>The value that Internet Explorer sends to Web servers to identify itself.</td> </tr> <tr> <td>Version Vector</td> <td>The value used in conditional commenting, which can also be used to affect CSS, markup and script blocks.</td> </tr> <tr> <td>Document Mode</td> <td>The value used to determine whether Internet Explorer uses the most recent behavior for CSS, DOM, and JavaScript operations, or emulates a previous version of Internet Explorer for compatibility.</td> </tr> </tbody> </table> <p>b. On the Developer Tools menu bar, the Browser Mode offers three options, each modifying the properties described in the preceding table in a different way. The following table describes these options.</p> <table border="1" data-bbox="508 1276 1414 1879"> <thead> <tr> <th>Browser Mode</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>Internet Explorer 7</td> <td>In this mode, Internet Explorer 8 reports a user agent, version vector, and document mode as if in Internet Explorer 7. Use this mode to test how Internet Explorer 7 users experience your site.</td> </tr> <tr> <td>Internet Explorer 8</td> <td>In this mode, Internet Explorer 8 reports a user agent, version vector, and document mode to match the default browser behavior, which is the most standards-compliant mode in Internet Explorer 8. Use this mode to test how Internet Explorer 8 users experience your Web site.</td> </tr> <tr> <td>Internet Explorer 8 Compatibility View</td> <td>In this mode, Internet Explorer 8 reports version vector and document mode as if in Internet Explorer 7, but</td> </tr> </tbody> </table>	Property	Description	User Agent string	The value that Internet Explorer sends to Web servers to identify itself.	Version Vector	The value used in conditional commenting, which can also be used to affect CSS, markup and script blocks.	Document Mode	The value used to determine whether Internet Explorer uses the most recent behavior for CSS, DOM, and JavaScript operations, or emulates a previous version of Internet Explorer for compatibility.	Browser Mode	Description	Internet Explorer 7	In this mode, Internet Explorer 8 reports a user agent, version vector, and document mode as if in Internet Explorer 7. Use this mode to test how Internet Explorer 7 users experience your site.	Internet Explorer 8	In this mode, Internet Explorer 8 reports a user agent, version vector, and document mode to match the default browser behavior, which is the most standards-compliant mode in Internet Explorer 8. Use this mode to test how Internet Explorer 8 users experience your Web site.	Internet Explorer 8 Compatibility View	In this mode, Internet Explorer 8 reports version vector and document mode as if in Internet Explorer 7, but
Property	Description																
User Agent string	The value that Internet Explorer sends to Web servers to identify itself.																
Version Vector	The value used in conditional commenting, which can also be used to affect CSS, markup and script blocks.																
Document Mode	The value used to determine whether Internet Explorer uses the most recent behavior for CSS, DOM, and JavaScript operations, or emulates a previous version of Internet Explorer for compatibility.																
Browser Mode	Description																
Internet Explorer 7	In this mode, Internet Explorer 8 reports a user agent, version vector, and document mode as if in Internet Explorer 7. Use this mode to test how Internet Explorer 7 users experience your site.																
Internet Explorer 8	In this mode, Internet Explorer 8 reports a user agent, version vector, and document mode to match the default browser behavior, which is the most standards-compliant mode in Internet Explorer 8. Use this mode to test how Internet Explorer 8 users experience your Web site.																
Internet Explorer 8 Compatibility View	In this mode, Internet Explorer 8 reports version vector and document mode as if in Internet Explorer 7, but																

Tasks	Detailed Steps									
		<p>the user agent string includes a token indicating that it is truly Internet Explorer 8 along with the Internet Explorer 7 string. Use this mode to test how Internet Explorer 8 users experience your Web site if they choose the Compatibility View option in Internet Explorer.</p>								
	<p>Note: Once a user clicks on the Compatibility View button , the browser saves your domain to a Compatibility View list. Subsequent visits to the domain will be automatically rendered under the Internet Explorer 8 Compatibility View mode. A user can Add or Remove domains from their browser's Compatibility View list by selecting Compatibility View Settings on the Tools menu.</p>									
<p>2. Understanding Document Modes</p>	<p>a. Internet Explorer 8 introduces the concept of document compatibility, which controls how the browser interprets and displays Web pages. The Developer Tools enable you to dynamically choose the compatibility mode used to display a Web page. You can thus easily see visual differences between various compatibility modes, and discover the most appropriate compatibility mode for a Web page.</p> <p>b. To change the document compatibility mode for a particular Web page, on the Developer Tools menu click Document Mode. This mode defines how Internet Explorer renders a Web page, but it has no effect on the version vector or user agent string. Using this option with the Browser Mode allows you to quickly determine which document mode is most suitable for your Web site. The Document Mode offers three options:</p> <table border="1" data-bbox="506 1136 1417 1787"> <thead> <tr> <th data-bbox="506 1136 971 1178">Document Mode</th> <th data-bbox="971 1136 1417 1178">Description</th> </tr> </thead> <tbody> <tr> <td data-bbox="506 1178 971 1478">Quirks</td> <td data-bbox="971 1178 1417 1478">When a document is rendered with no document type or a quirks document type, this behavior matches that of Internet Explorer. It is similar to the behavior of Microsoft Internet Explorer 5 and the quirks mode behavior of Internet Explorer 6, and it is the same as the quirks mode behavior of Internet Explorer 7.</td> </tr> <tr> <td data-bbox="506 1478 971 1583">Internet Explorer 7 Standards</td> <td data-bbox="971 1478 1417 1583">This behavior matches that of Internet Explorer 7 when it renders a document that has a strict or unknown type.</td> </tr> <tr> <td data-bbox="506 1583 971 1787">Internet Explorer 8 Standards</td> <td data-bbox="971 1583 1417 1787">This is the latest standards-compliant behavior available in Internet Explorer 8 and is the default mode used by Internet Explorer 8 to render documents that have a strict or unknown document type.</td> </tr> </tbody> </table>		Document Mode	Description	Quirks	When a document is rendered with no document type or a quirks document type, this behavior matches that of Internet Explorer. It is similar to the behavior of Microsoft Internet Explorer 5 and the quirks mode behavior of Internet Explorer 6, and it is the same as the quirks mode behavior of Internet Explorer 7.	Internet Explorer 7 Standards	This behavior matches that of Internet Explorer 7 when it renders a document that has a strict or unknown type.	Internet Explorer 8 Standards	This is the latest standards-compliant behavior available in Internet Explorer 8 and is the default mode used by Internet Explorer 8 to render documents that have a strict or unknown document type.
Document Mode	Description									
Quirks	When a document is rendered with no document type or a quirks document type, this behavior matches that of Internet Explorer. It is similar to the behavior of Microsoft Internet Explorer 5 and the quirks mode behavior of Internet Explorer 6, and it is the same as the quirks mode behavior of Internet Explorer 7.									
Internet Explorer 7 Standards	This behavior matches that of Internet Explorer 7 when it renders a document that has a strict or unknown type.									
Internet Explorer 8 Standards	This is the latest standards-compliant behavior available in Internet Explorer 8 and is the default mode used by Internet Explorer 8 to render documents that have a strict or unknown document type.									

Tasks	Detailed Steps
	<p>Note: Similar to changes in other developer tools, changing the Browser Mode or the Document Mode is temporary and does not affect the underlying source of a Web page. A selected mode will remain in effect for a browser tab until another mode is chosen or the tab is closed.</p>
<p>3. Testing Browser Modes</p>	<ul style="list-style-type: none"> a. With the Northwind Traders product page debugged we can now test to see what it will look like in other versions of IE. b. Open debugging.html in IE8 (if not already open) and press F12 to open the Developer Tools pane. Click "Browser Mode: IE8" on the toolbar to drop down other browser mode choices. Select "Internet Explorer 7" and watch the page for any changes. c. The only things that changed were the product title and a few spacing issues. There are no major problems as far as compatibility with IE7 is concerned.
<p>4. Testing Document Modes</p>	<ul style="list-style-type: none"> a. With your Browser Mode set to IE8 click "Document Mode: IE8 Standards" to view the other options. Choose "Quirks Mode." You will now see that the image block is moved to the bottom of the Description and Key Features text. This simulates how this webpage would look in IE7 in Quirks Mode. This is useful for seeing the differences between Quirks Mode and Standards Mode rendering. Most often, you will use this to test how a Quirks Mode page will render in Standards Mode. b. This concludes the Debugging and Application Compatibility Lab.

Related Resources:

[Internet Explorer 9 Beta Download](#)