

# Using *HTML Kit* to Debug Server-Side Scripts

by  
Peter Lavin

April 21, 2004

## **Overview**

Browsers are notorious for rendering HTML pages perfectly even when a page contains errors. For instance, a button may display properly but not function because of a missing closing tag. Hence the need for tools to check syntax. This is relatively straightforward for static HTML pages but what about dynamic pages? When server-side technologies such as PHP or ASP are used to create HTML pages dynamically, checking and debugging the result can be difficult.

This article will show how this can be done using the free text editor, *HTML-Kit*. This will involve configuring some of the features of this editor. No knowledge of any specific technologies is required but a general knowledge of HTML and of the function of browsers and web servers is assumed.

## ***Introduction***

I do not intend to repeat anything covered in my introductory article on *HTML-Kit* except to note a few features that directly relate to the current discussion. What I intend to do here is explain some of the ways this application may be used to simplify the writing, validation and debugging of server-side scripts. If you need to download and install *HTML-Kit* please see the instructions at <http://www.chami.com/htmlkit/>.

To check your server-side scripts - and it is always good practice to check them locally before uploading them - you need a server. Since *HTML-Kit* only runs under Windows you will most likely be using Personal Web Server (PWS) or Internet Information Server (IIS). All versions of Windows after Windows 95, excepting ME, support running these web servers locally. If you are adventurous and have Installed Apache Web Server under Windows this will also work and you should be able to follow along making the appropriate changes.

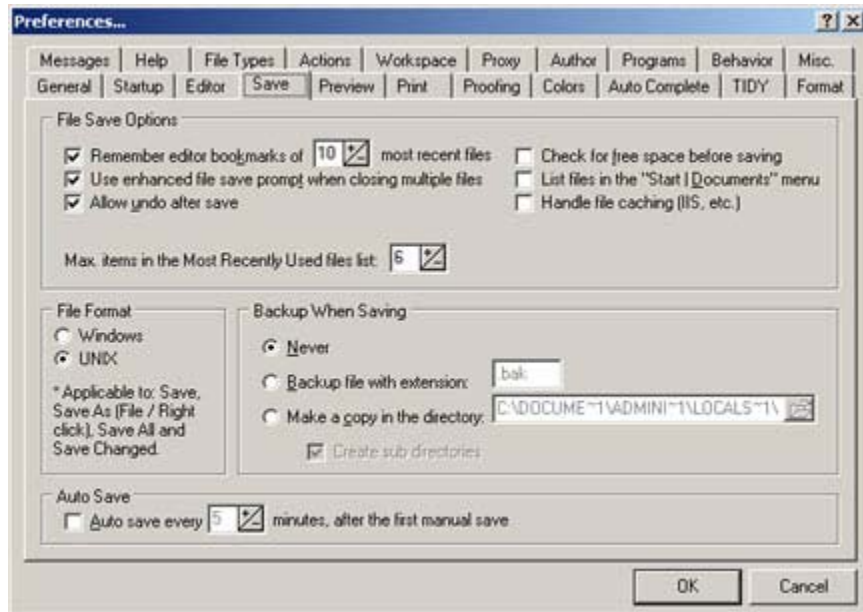
Naturally enough, all versions of PWS and IIS support server-side scripting using ASP but, as mentioned above, what's detailed here applies to any server-side script supported by your server.

## ***Getting Started***

Since your files will be processed by a web server make sure that you are working in a directory located in the home directory of your server. If you are using IIS or PWS then this is most likely "C:\inetpub\wwwroot". As long as your directory is under the root directory of your server you do not need to turn on web sharing. However, it is not a bad idea to do so. This can be done through IIS or PWS server administration or simply by right clicking the appropriate directory and choosing the menu option "Properties" and then the "Web Sharing" tab. If there is no "Web Sharing" tab then make sure that your server is installed and is running.

## ***Configuring HTML-Kit***

In order to use this HTML editor to debug server-side scripts we need to edit the user preferences. This can be rather intimidating as there are over twenty tabs in the configuration window and many options on each page. Have a look. Press the "Edit" menu option and then, at the very bottom, choose "Preferences". You should find a screen similar to the one below.



In this article we will only be dealing with a very limited number of settings. These will be, firstly, some basic settings to increase usability, secondly, telling *HTML-Kit* where your server is located and finally, making file associations.

## Basics

Something that you might want to pay immediate attention to is the file format setting on the “Save” tab of the preferences window. If you know that the server you are planning to upload your files to runs on a Linux or Unix platform then be sure to choose the Unix file format. The reason for this is that different operating systems handle the return key in different ways. If you save your files in Windows format and later download them from a Unix server you will find that an extra carriage return has been inserted after each line. While this does not change your file’s functionality it can be annoying. As an additional note, having saved your HTML file in Unix format, it will not display properly in Notepad. If you want to view it in a Windows text editor use WordPad instead. If you do save it from WordPad make sure you save it in text format.

I’ll just point out that you can quickly access commonly performed tasks by adding them to “Favorites”. If you right click a button on any one of the elements in the Actions Bar (the tabbed strip below the tool bar) a pop-up menu allows you to add this item to your favourites. When you become familiar with *HTML-Kit* and find that you repeatedly use certain capabilities you will really appreciate this feature.

Exploring the other settings will certainly improve your efficiency with this tool but these are really the only settings you may need to worry about here.

## Set Up the Server

*Html-Kit* needs to know where your server is located if it is going to process your server-side scripts. Again, this is configured from the “Preferences” menu option so open the preferences window and choose the “Preview” tab. In the frame entitled “Preview Rules” make sure that the “Enable Server Mappings” box is checked. Having done that, click the “Edit Preview Rules” button. Here you will add a server mapping. Clicking the “Add” button will open a dialogue window and you will be able to add a File Path and a server address. If you are running IIS or PWS on your local machine the file path will be “C:\inetpub\wwwroot\”. For the server address put “http://<your computer name>/” substituting the appropriate value for “your computer name” or, if you like, you may use “localhost”. Make sure you click “Okay” to save your preferences. Now, whenever you preview files that are in a directory at or below the root directory of your server, they will be processed by your server. If your server-side script generates HTML the server will create it when you preview a file.

## File Associations

From the preferences window choose the “Programs” tab. In the “Windows Integration” frame click the button with the caption “File Associations, Explorer ...”. For the “Internet Explorer (IE) 5+ Integration” option make sure that the “Use to View Source” box is checked. This will ensure that when we preview a file in *HTML-Kit* and then choose to view the source code of a page it will also be shown in *HTML-Kit*. Since we’ve already told *HTML Kit* to run this file through our local server, any server-side scripting will have been processed and output as HTML.

## View Source and Validate

As mentioned in my earlier article, one of the nicest features of *HTML-Kit* is the ability to preview your files. This capability is central to the current discussion. Briefly, to preview the current file all you need to do is click the “Preview” tab at the bottom of the “Editing Window”. The application defaults to showing how your page will look in Internet Explorer.

Having configured *HTML-Kit*, we can now deal with the problem of validating files that need to be processed by the server. Open any file that dynamically generates HTML and then press the “Preview” tab. When your processed page

shows, right click the preview window, avoiding any graphics, and then choose “View Source” from the pop-up menu. Doing this will open another window in *HTML-Kit*, a window that holds the HTML code as generated by your server.

A word of warning – make sure that you are previewing in Explorer Mode and not Gecko Mode. The preferences menu only allows you to view the source in Internet Explorer. If you are in Gecko mode this option will be disabled.

If your file has been properly processed by the server you are now in a position to verify your dynamically created web page. *HTML-Kit* has the capability of checking HTML code by uploading it to an online mark-up validation service. By choosing the menu options “Actions” and then “Online” you will have two choices for validating your HTML file. You may use the Web Design Group (WDG) or the World Wide Web Consortium (W3C) validator. I prefer to use the W3C validator because this is the organisation responsible for developing guidelines and standards for the web. Whichever service you choose, your file will be uploaded and a report generated. Any coding errors that exist will be referenced by line number. You can even print out your code with all the errors referenced. The W3C service will also inform you of any deviations from the standard - if you are missing the “alt” attribute from an “img” tag for instance. While these errors will not affect the functionality of your page, why not write code to conform to the standard? It will certainly improve the interoperability of your page and make it more user friendly.

## ***Conclusion***

Through a few rather simple steps *HTML-Kit* can be set up to validate HTML code generated by a server-side script. Dynamically created pages may be viewed in *HTML-Kit* and the resultant HTML page uploaded to a validation service to check its syntax. All this functionality integrated into one tool makes it an easy way to debug server-side scripts.

## ***About the Author***

Peter Lavin runs a Web Design/Development firm in Toronto, Canada. For more information visit <http://www.softcoded.com/> . He may be reached at [peterlavin@sympatico.ca](mailto:peterlavin@sympatico.ca) .