

HTML PowerAnalyzer Help Contents

Getting Started

- [Introduction](#)
- [Quick Start](#)
- [About the Main Window](#)

How to

- [Define a Website Project](#)
- [Select Options](#)
- [Run an Analysis](#)
- [Run an Analysis in Silent Mode](#)
- [Interpret the HTML Errors Reported](#)
- [Customize the Reports](#)

Other Stuff

- [About HTML PowerAnalyzer](#)
- [Other HTML PowerTools](#)



<http://www.tali.com>

sales@tali.com

support@tali.com

Introduction

HTML PowerAnalyzer is a sophisticated tool employing powerful algorithms to scan HTML files and alert the user to all errors contained within them. In addition, a comprehensive report is generated containing a wealth of useful information about each file, and the entire Website.

In addition to all types of HTML syntax errors, HTML PowerAnalyzer will catch invalid &character; entity codes, non-text characters, missing/invalid link references (i.e., files pointed to by HREF, SRC, etc.), missing anchors, and link references containing capital letters(which may be cause problems on case-sensitive Unix servers, for example). (In this version, all files contained in the Website must reside on the user's hard disk. Planned for a future release is the ability to scan a Website stored on a remote server and perform all scanning and link validation remotely.)

In addition, HTML PowerAnalyzer builds a list of all files included in the project directories that are not included in the Web project and that are not referenced by any files in the Web project. This helps you weed out old and obsolete files that may still be taking up space unnecessarily.

HTML PowerAnalyzer has an additional feature useful when translating Web pages into other languages, or when allowing third parties to edit the text contained in HTML files. It can generate files containing only the HTML tags found in translated HTML files, which can then be compared with the original using a word processor or text editor with the capability. This can be used to make sure that no changes were made to the HTML tags themselves during the process.

HTML PowerAnalyzer supports the very latest HTML 4, Netscape 4 extensions, and Microsoft Internet Explorer 4 extensions, and can be completely customized. The user can even select which browser (or HTML standard) to analyze for: any proprietary HTML tags (or parameters within standard tags) not supported by the selected browser will be flagged.

HTML PowerAnalyzer's algorithms utilize databases containing all rules of the HTML language -- an [HTML Rulebase](#). The [HTML Rulebase Editor](#), available separately, allows the advanced HTML user to freely modify any and all aspects of the HTML markup language for his particular purposes. This includes adding/deleting HTML tags and tag parameters, redefining language rules for all defined elements, adding/deleting browser definitions, and more. In other words, the user has the ability to totally customize the logic used in the analysis. Another advantage of the HTML Rulebase is that it can be updated by downloading up-to-date files from Talicom's website, www.tali.com, as they become available. Thus, no matter how quickly the HTML language evolves, HTML PowerAnalyzer will never become obsolete. Finally, the HTML Rulebase Editor is a valuable online reference to every HTML tag and its attributes.

In today's rapidly changing WWW landscape, it is not enough to visually check a Website in one or two browsers -- you want to be certain that your markup is perfect and error-free. You also want to know for sure that every single image and hyperlink is perfect, without having to scour your site and test-click every link. HTML PowerAnalyzer will automatically provide you with the certainty you need -- with the click of a button.

See also:

[Quick Start](#)

[Defining a Website Project](#)

[Running an Analysis](#)

Quick Start

Step 1: Select a Website project to analyze

1. After starting the program, click [Select Project](#).
2. Click New to define a new Website project to be analyzed.
3. Enter the required [details](#). Click Help for information on any item.
4. Click OK.
5. Click Select.

Step 2: Select the options you wish to use

1. Click [Options](#).
2. Select an HTML Rulebase to Use from the list (in the Analysis tab).
3. Select a report template to use (in the Report tab).
4. Choose a directory to receive the report (in the Report tab).
5. Set the other options as you wish. Click Help for more information on any item.
6. Click OK.

Step 3: Run the analysis

1. Click Analyze HTML.
2. Confirm that the file list shown is what you intend to analyze.
3. Click Proceed.
4. To see the analysis results of each file, make sure Always Pause Between Files is checked. You will then be able to click [View File Results](#) between files. Alternatively, you can choose to have the program pause only if an error occurred -- see Options for more information.
5. When the analysis is complete, click Launch Report Viewer to see the comprehensive report.

Note: After initially defining Website projects and selecting options, you can run HTML PowerAnalyzer in Silent Mode by either defining command line parameters or by dragging a single HTML file onto the HTML PowerAnalyzer icon. See [Silent Mode](#) for more information.

Defining a Website Project

Before performing an analysis, you must define the Website to be analyzed. You do this by clicking [Select Project](#) in the main window, then New in the Project Manager window. After entering the required [details](#) that define a project, click OK, then Select.

You can now perform an analysis on all the files contained in your defined project.

You can define any number of separate projects. To select a project to be analyzed, simply select it in the [Project Manager window](#) and click Select.

Note: To analyze a single file, that is part of a defined project or not, you can click Other in the [Project Window](#).

Options

All options are set in the HTML PowerAnalyzer Options window, accessible by clicking Options from the [main window](#).

The window is divided into three categories: [Analysis](#), [Report](#), and [Program](#).

Analysis

- Check internal links - When this option is selected, all local links (links to a file or anchor within a file in the current project, also known as relative references) will be validated. If an indicated file or anchor is not found, it will be reported as such.
- Honor <BASE> tag - The optional HTML tag BASE is used to indicate that all local (relative) references in the file should be relative to the URL specified in the BASE tag as opposed to the location of the current file. To determine actual file locations of relative references according to the BASE tag, when found, select this option.
- Include warnings in report - There are a number of [errors](#) that can be included in the report, that are more warning notes than actual errors. For example, when a tag's attribute value may be missing (e.g.,) or when an SGML character (such as & or >) is found in the file. If you wish to see these notes, select this option. To ignore them, leave this option unselected.
- Report non-text characters - Officially, HTML may only include characters specified in ISO-8859-1, aka US-ASCII, aka Latin-1. To ensure consistent display of HTML documents on any platform, the use of other characters is discouraged. HTML PowerAnalyzer takes a strict approach regarding non-text characters in a file, and, if this option is selected, will report any character not the ASCII range of 32 through 126 plus 9 (tab). If this option is not selected, no error reports will be made.
- Non-text character threshold - The number entered here determines when the analysis on a given file will halt upon finding non-text characters (see Report Non-text Characters, above, for details). Setting this option to a relatively low number (such as 10) will save time when you've accidentally specified some non-HTML file to be analyzed. If a binary file, for example, is being analyzed, the analysis will stop right away. If, on the other hand, you choose to use non-text characters in your HTML documents, set this to a high value (32,000 is the maximum) to avoid halting an analysis. If set to zero, the analysis will halt when the first non-text character is encountered in the file.
- Create line-numbered files - When this is checked, a separate file, with the extension .NUM, is created for each file analyzed. The .NUM file is simply a copy of the original HTML file with line numbers added. This is a convenience for those situations when your editor doesn't display the line number of the cursor, or when HTML PowerAnalyzer's line numbering (as indicated in error messages) doesn't agree with that of your editor. This latter situation can occur when non-standard line break codes are contained in the original HTML file, for example CR CR LF instead of the standard CR LF (ASCII 13 10).
- Create tag log files - When this is checked, a separate file, with the extension .TAG, is created for each file analyzed. The .TAG file contains a copy of every HTML tag found in the analyzed, one per line. These files are useful when translating Web pages into other languages, or when allowing third parties to edit the text contained in HTML files. Using a word processor or text editor with the capability, you can compare the .TAG files of the

original and the translated version to make sure that no changes were made to the HTML tags themselves during the process. Before comparing, you might want to use a search & replace function to eliminate those tags or tag attributes that may be legally changed, such as <P>,
, and the image tag's ALT. Note that all comment tags are represented simply as <!-- -->.

- HTML Rulebase to Use - Select from the list the browser or HTML specification with which to analyze your files. For example, if you want to make sure that your files contain only HTML tags recognized by Netscape 2, then select Netscape 2 from the list. The choices in this list represent the available [HTML Rulebase](#) files you have in the program directory. To add one to the list, simply copy its Rulebase file to the program directory. Updated Rulebase files are available at www.tali.com.
- Customize - [HTML Rulebases](#) are fully customizable using the [HTML Rulebase Editor](#). Clicking the Customize button here launches the HTML Rulebase Editor with the currently selected Rulebase open for editing.

Report

- Output directory - Enter the path where you want the analysis reports to be saved. Use the Browse button to select an existing directory (recommended to avoid typos). Note that no validation of the specified path is made here; if the path is invalid an error will occur during the analysis.
- Template to use - Select which report template you wish to use to generate the analysis report. A number of reports templates are supplied with the program, and you can create your own as well. The information displayed below the list tells you a bit about the template. See [Customize Report Template](#) for details on the meaning of these settings.
- Customize - To modify the selected report template, simply click this button. The [Customize Report Template](#) window will appear.
- Delete - To permanently erase a report template, click this button.

Program

- Always confirm file list - Each time you run the program or select a new project, the file list to be analyzed (based on the current project's definition) must be refreshed (either by clicking Refresh File List or by performing an analysis). If this option is selected, you will be asked if you want to refresh to file list every time you click the Analyze HTML button. If it is not selected, the file list will be automatically refreshed once, and then not again. If you will be making changes to the files contained in your site in between analyses, or if you want to exclude certain files for different analyses, select this option. Otherwise, it is less irritating to leave this option unselected.
- Copy report filename to clipboard - This is a convenience option to more quickly load the report generated at the conclusion of an analysis. When selected, you can go to the File, Open window of your browser or text editor and simply paste in the report's file name instead of typing it in. Leave this option unselected if you wish to avoid clearing the Windows Clipboard after each analysis.
- Always pause between files - When selected, the analysis will pause after each file giving you the chance to view each file's results (HTML errors and analysis of file/anchor references) before continuing to the next file. If you want to make changes based on the

analysis one file at a time, select this option. To complete an entire analysis without user interaction and then view all the information in the report after the analysis is complete, leave this option unselected. When this option is selected, the two options below it are disabled as they are then redundant.

- Pause after files with HTML errors - Only available when Always Pause Between Files is not selected. When selected, the analysis will pause after each file containing one or more HTML errors giving you the chance to view the file's results before continuing with the next file. If the file contained no errors, the analysis would automatically continue with the next file. To complete an entire analysis without user interaction and then view all the information in the report after the analysis is complete, leave this option unselected.
- Pause after files with invalid links - Only available when Always Pause Between Files is not selected. When selected, the analysis will pause after each file containing one or more invalid references giving you the chance to view the file's results before continuing with the next file. If the file contained no invalid references, the analysis would automatically continue with the next file. To complete an entire analysis without user interaction and then view all the information in the report after the analysis is complete, leave this option unselected.

Running an Analysis

After [defining and selecting a project](#) and [setting preferred options](#), click Analyze HTML in the [main window](#).

If the file list has not yet been refreshed, it will be automatically refreshed now. If it has already been refreshed and the option Always Confirm File List is selected, you will be given the chance to refresh it again. You can always manually refresh the file list before an analysis by clicking Refresh File List in the main window.

After refreshing the file list, you will either see the list of selected files about to be analyzed, or the name of the home page from where the project analysis will start. The latter is the case when Only scan linked files is selected in the [project definition](#) - click Proceed (or OK) to continue. When the former is the case, you have four choices:

- You may click Proceed to accept the list and perform the analysis. (If you clicked the Refresh File List button, the Proceed button will instead be OK).
- You may change the permanent definition of the project. To do this, click [Modify Project](#) and make any desired changes. Note that this is the same as clicking Properties in the Project Manager window and that all changes are saved. When done, click Proceed.
- You may choose to exclude certain files from the current analysis. To do this, click [Exclude Files](#) and use the four Include/Exclude buttons to make your selection. Note that this selection will only affect the current analysis and will not be saved. When done, click Proceed.
- You may disregard the currently defined project and select a single file to analyze. To do this, click the Other button and select a file. When done, click Proceed. Note that this has no impact on the project definition, and is a one-time selection. If you do not click Refresh File List, however, you can re-run the analysis on the selected file as many times as you wish without having to reselect it.

The [analysis procedure](#) will now begin.

Note: You can also run HTML PowerAnalyzer in Silent Mode by either defining command line parameters or by dragging a single HTML file onto the HTML PowerAnalyzer icon. See [Silent Mode](#) for more information.

Understanding the HTML Errors Reported

- (0) Unrecognized tag. The HTML tag does not exist in the currently selected [HTML Rulebase](#). Check the spelling of the tag for a typo, and make sure that it is followed immediately by a space.
- (1) Non-text character in file. A character was found that was not in the ASCII range of 32 through 126. For more information, and to disable reports of non-text characters, see [Setting Options](#).
- (2) Missing closing bracket. This occurs when a tag was started (i.e., following a < character), but the end of the file was reached before it was closed (i.e., no > character was found). If you cannot find the cause, search the file for a < character in the file that is not intended to be the start of a tag. If you need to insert this character in a file, use the < glyph code instead.
- (3) Last character in file is <. An HTML file should end with </HTML>. In this case, the last character in the file is a < character.
- (4) Analysis aborted: too many non-text characters in file. NOTE: All information about this file in this report must be deemed incorrect. Based on the Non-text Character Threshold value in [Options](#), the analysis was halted in the middle. As a result, reported information about the file (including the project statistics reported) will probably be incorrect. This generally occurs when a non-HTML file has been subjected to analysis. If in fact it is an HTML file in which you use many non-text characters, set the Non-text Character Threshold to a large number (see [Setting Options](#) for more detail on this). Whenever an analysis is interrupted it is likely that parts of the report are incorrect. Do not trust any report that did not complete successfully.
- (5) File: Missing close tag. This indicates that the end of the file was reached and there are one or more a mismatched open tags of a tag pair (if there are more than one mismatch, the number of occurrences will be indicated). In other words, there are more open tags than close tags in the file of the specified HTML tag. This error may be redundant if an open tag followed a previous one that was not yet closed. For example, a was found in the file and before closing the bold block with a another was found (or the end of the file was reached first), an error would have already been reported. Find the open tag in the file without a paired close tag.
- (6) Invalid space after open tag bracket. The < character in an HTML file always indicates the start of an HTML tag. The tag must immediately follow the < character. This error occurs when a < followed by a space is found.
- (7) Invalid empty tag. The invalid sequence <> was found in the file.
- (8) Close tag with no previous open tag. This indicates that the closing tag of a tag pair was found but there was no preceding open tag. For example, a tag was found but there was no preceding tag. This error will also be generated when there was a previous tag but another tag had already been found. This type of error can be automatically be fixed by the separate HTML PowerTool, [HTML Tag Pair Fixer](#).
- (9) Tag must appear inside ___ tag pair. According to the currently selected [HTML Rulebase](#), this HTML tag is only permitted inside a pair of the indicated HTML tag. For example, the <TR> tag is only valid within the <TABLE> ... </TABLE> tag pair.
- (10) Tag's maximum number of occurrences (__) exceeded. According to the currently

selected [HTML Rulebase](#), this HTML tag is only permitted a certain number of times (as indicated in the error message) in a single HTML file. This tag occurrence has exceeded that limit. For example, the <TITLE> tag is only valid once per file.

- (11) EOF reached before closing tag. This error occurs when a tag defined in the currently selected [HTML Rulebase](#) as "ignore contents" is opened, but not closed before the end of the file. An example of this is the comment tag, <!--. This will occur if, for example, the last line of the file was, <!-- this is the last line of the file. The error indicates in this example that there was no --> found.
- (12) File: Required tag not found in file. According to the currently selected [HTML Rulebase](#), the indicated tag **must** appear in every HTML file, yet it was not found in this file. For example, there was no <HTML> tag in the file. An example of a tag officially required by every HTML file that is not universally used in practice is <!--DOCTYPE...>. If you do not wish to include this tag and you do not want to see this error every time you analyze your files, you can customize its "required" definition using the the [HTML Rulebase Editor](#).
- (13) Invalid open tag before closing previous pair. The open tag of a tag pair was found, but a previous occurrence of the same open tag was never paired with a close tag. For example, bold text was found. Note that in many cases the error actually exists **before** the line where this error is reported, since the missing Close tag is usually missing after a previous instance of this Open tag. Note also that if the tag pair is defined as Nestable in the currently selected [HTML Rulebase](#), this error will never occur for that tag. is an example of a "nestable" tag -- you may have one ordered list inside another.
- (14) Must close the __ dependant tag pair(s) before closing this one. This error is the complement of error 9. This error occurs when a tag pair that must, according to the currently selected [HTML Rulebase](#), exist only within another tag pair is still open (i.e., an open tag has been found, but not its closing tag) and the "parent's" close tag has been found. For example, <HEAD> <TITLE>Title Text </HEAD>. In this example, the <TITLE> ... </TITLE> tag pair is still open but the closing </HEAD> tag has been found. If there are more than one "dependent" (or "child") tag pairs still open, the number will be indicated in the error message.
- (37) Tag may not appear inside a table cell. According to the currently selected [HTML Rulebase](#), it is illegal to use the specified tag inside a table cell (e.g., between <TD> ... </TD> tags). Note: table behavior for tags is defined in the Advanced section when editing a tag in the HTML Rulebase Editor.
- (38) Missing close tag in table cell. This indicates that the end of the current table cell was reached and there are one or more a mismatched open tags of a tag pair. In other words, there are more open tags than close tags in the table cell of the specified HTML tag. The following example would cause the tag to be flagged as an error since the cell was closed, but the bold tag was not closed inside the cell in which it was opened: <TD>bold text</TD>.
- (50) Unexpected quote mark in parameter value. An incorrectly-placed quotation mark " was found in a parameter value. For example, . In this example, the ALT parameter must either be specified as ALT=text or ALT="text".
- (51) Missing closing quote mark in parameter value. A parameter value specification begins with a quotation mark, but a closing quotation mark was not found. For example, in <IMG ALT="text" there is no closing " after the value "text".

- (52) Invalid use of quote marks in parameter value. In specifying parameter values, quotation marks can only be used to delimit a string, as in . Any other use is invalid.
- (53) All parameters invalid for tag. According to the currently selected [HTML Rulebase](#), the specified HTML tag takes no parameters, yet the tag contains one or more parameters.
- (54) Parameter ___ not valid for this tag. According to the currently selected [HTML Rulebase](#), the specified parameter is not supported by the specified HTML tag.
- (55) Parameter value may be invalid for ___. Unless the specified parameter is allowed take a null string (e.g., "") as a value, the parameter's value is invalid.
- (56) Required parameter ___ missing. According to the currently selected [HTML Rulebase](#), the specified parameter **must** appear in the specified HTML tag.
- (57) Neither required parameter ___ nor ___ present. According to the currently selected [HTML Rulebase](#), at least one of the two specified parameters **must** appear in the specified HTML tag, yet neither was found.
- (58) Missing reference parameter value for ___. According to the currently selected [HTML Rulebase](#), the specified parameter of the specified tag must contain a valid reference to a file or anchor, yet it does not.
- (70) Invalid SGML character: lone >. An unexpected > character was found in the file. If you need to insert this character in a file, use the > glyph code. (Note that an error does not exist for the < character, since this is always considered to be the beginning of an HTML tag -- unless in a section defined as 'ignore inside' such as inside <!-- comment -->.)
- (71) Invalid SGML character: lone &. An unexpected & character was found in the file. If you need to insert this character in a file, use the & glyph code.
- (72) Undefined character entity. The specified &character; entity code is not defined in the currently selected [HTML Rulebase](#).
- (99) Analysis interrupted by user or error. The placement of this message indicates at what point the analysis was interrupted, but not the reason why. Whenever an analysis is interrupted it is likely that parts of the report are incorrect. Do not trust any report that did not complete successfully!

Other HTML PowerTools

The HTML tools listed on this page are available from Talicom(R). All run in the Windows environment. Please visit our home page at www.tali.com for more information.

HTML PowerAnalyzer

HTML PowerAnalyzer is a sophisticated tool employing powerful algorithms to scan HTML files and alert the user to all errors contained within them. In addition, a comprehensive report is generated containing a wealth of useful information about each file, and the entire Website.

In addition to all types of HTML syntax errors, HTML PowerAnalyzer will catch invalid &character; entity codes, non-text characters, missing/invalid link references (i.e., files pointed to by HREF, SRC, etc.), missing anchors, and link references containing capital letters (which may be cause problems on case-sensitive Unix servers).

In addition, HTML PowerAnalyzer builds a list of all files included in the project directories that are not included in the Web project and that are not referenced by any files in the Web project. This helps you weed out old and obsolete files that may still be taking up space unnecessarily.

HTML PowerAnalyzer supports the very latest HTML 4, Netscape 4 extensions, and Microsoft Internet Explorer 4 extensions, and can be completely customized. The user can even select which browser (or HTML standard) to analyze for: any proprietary HTML tags (or parameters within standard tags) not supported by the selected browser will be flagged.

HTML PowerAnalyzer's algorithms utilize databases containing all rules of the HTML language -- an [HTML Rulebase](#). The HTML Rulebase Editor allows the advanced HTML user to freely modify any and all aspects of the HTML markup language for his particular purposes. This includes adding/deleting HTML tags and tag parameters, redefining language rules for all defined elements, adding/deleting browser definitions, and more. In other words, the user has the ability to totally customize the logic used in the analysis. Another advantage of the HTML Rulebase is that it can be updated by downloading up-to-date files from Talicom's website, www.tali.com, as they become available. Thus, no matter how quickly the HTML language evolves, HTML PowerAnalyzer will never become obsolete.

In today's rapidly changing WWW landscape, it is not enough to visually check a Website in one or two browsers -- you want to be certain that your markup is perfect and error-free. You also want to know for sure that every single image and hyperlink is perfect, without having to scour your site and test-click every link. HTML PowerAnalyzer will automatically provide you with the certainty you need -- with the click of a button.

HTML Rulebase Editor

A great strength of HTML PowerAnalyzer (as well as some other HTML PowerTools) lies in the customizable [HTML Rulebase](#) files that contain the rules of the HTML markup language. Due to the many different implementations of HTML in the real world, and the rapid pace at which the language is presently evolving, it is an absolute necessity to be able to quickly and easily customize any software dealing with HTML.

The HTML Rulebase Editor allows you to do just that. No matter how quickly the vendors of HTML editors and other HTML programs react to changes in the language, they will never

keep up. But you, the user, will always want to be at the forefront. Using the HTML Rulebase Editor, the HTML PowerTools toolkit can always be completely up-to-date.

The HTML Rulebase Editor features a professionally-designed user interface to allow you intuitive and direct access to every relevant attribute of every HTML tag and tag parameter. You can define all aspects of tags and tag parameters for each specific browser (or HTML standard), and even add support for brand new browsers. For example, when Netscape Navigator 5 is released, you can immediately enter all of its new commands and specify them as valid only for that particular browser.

In addition to tag and parameter information, you can modify the lists of defined protocols (e.g., http://, ftp://) and character entity codes (e.g., & ©);).

The HTML Rulebase Editor is also an excellent online reference to the entire HTML language. Brief descriptions are included for every tag and tag attribute - and you can add your own.

The flexibility and power that the HTML Rulebase Editor provides for users of HTML PowerAnalyzer are unmatched in the HTML software available on the market today.

HTML Meta Manager

HTML Meta Manager is the fastest, easiest, and cheapest way to guarantee that your Website appears in every major WWW search engine. It allows rapid insertion/editing of Description and Keyword META tags (and TITLE tags) for every page in a Website.

The major WWW search engines, including Alta Vista, Lycos, Infoseek, and WebCrawler, constantly scan the World Wide Web to automatically index every page they find -- including yours. In the absence of any special indicators as to the content of your page, they take a best guess at an accurate description and applicable search keywords. The result is often less than satisfactory, which is why (a) so many searches turn up garbage, and (b) why your site might not come up when someone is searching for it.

So what kind of special indicators can you use to improve the indexing of your Webpages in the search engines? Well, they're called META tags and they can be inserted into every HTML page in a Website. META tags explicitly define a description and keywords for every page in a Website. It is very much in your best interest to include these tags in your pages, if you want to guarantee that your site will come up when a potential visitor is using a search engine. And not just on your home page -- why not have every page in your Website come up separately in a search, improving the chances that someone will click on one of your pages, rather than the competition?

The problem is, adding the required HTML tags to every page in a Website can be a huge job. Some HTML editors (such as Netscape Navigator Gold 3) allow you to define a description and keywords while working on a page, but you still have to manually enter the information for every page separately. The result is that most Websites still do not have the required META tags entered on every page.

HTML Meta Manager is an elegant solution to this problem: it allows you to easily enter a description and keywords for every page in your Website in a single, easy-to-use window. You can enter separate information for each page, or add the same META information to every page in your Website with the click of a button. The program also allows you to easily edit each page TITLE, or to automatically insert the TITLE as the description for every page.

Regardless of the HTML editing environment you work in, HTML Meta Manager can quickly

and easily ensure that all your Web pages are properly listed in the major search engines.

HTML PowerSpell

HTML PowerSpell will allow you to quickly and easily spell-check entire Websites, regardless of what software you used to create them. This program is unique in its comprehensive understanding of HTML files to ensure that you spell-check everything you should while avoiding all the HTML code that you don't want to check.

Unlike the spell-checker that may be built into your HTML editor, you no longer have to laboriously check your work page-by-page. And unlike the spell-checker in your text editor, you no longer have to repeatedly hit Ignore to skip all the HTML elements that you don't want to check.

Advanced spell-checking features that you've come to expect from your software are all included, such as support for custom dictionaries, various options for customizing spell-checking, a context view of misspelled words, and support for multiple languages.

HTML PowerSpell fully understands your HTML files in order to thoroughly and correctly spell-check them. For example:

**<i>un</i>believable really means unbelievable
resumé; really means resumé
 should include Check this text
etc.**

Typos and misspelled words in your Websites may leave doubt as to your level of professionalism or your attention to detail. Use HTML PowerSpell every time you make changes to your Websites to ensure that your pages are error-free!

HTML PowerSearch

Find and replace utilities, included in editors and word processors as well as stand-alone tools, abound. So why is HTML PowerSearch better?

First of all, HTML files are not text files, even though they are saved as text. HTML files follow a specific set of rules in how their content is read by an HTML browser, and standard find and replace tools do not take this into consideration. For example, in HTML a space, a tab, and a line break are all equivalent. Well-formatted HTML source that is easy to read and work with contains many tabs and line breaks that will never be rendered when the file is viewed in a browser. Your standard find tool will not know to find the search string "hello world" in the following example, yet in HTML it should be found.

In this example we have the text "hello world" separated by a line break and a tab.

HTML PowerSearch also recognizes &entity; codes and that tags can appear in the middle of search text, so finding "a great day" is easy when the HTML source shows:

I'm having a great day.

Try that with your text editor or word processor! HTML PowerSearch is an HTML-specific tool that knows how to intelligently perform searches on HTML files.

Secondly, performing a search or search & replace across an entire Website using an editor or word processor can be very tedious. HTML PowerSearch handles entire Websites, stored in any number of subdirectories, with one button click.

Thirdly, HTML PowerSearch combines its HTML-specific searching with flexible wildcard searching. You will never be able to return to working on your Websites without the aid of HTML PowerSearch.

HTML Image Scanner

Experienced Web developers know -- and beginners will learn -- the value of using the WIDTH and HEIGHT parameters of the IMG tag: much faster perceived loading of a Web page. When the browser is provided with these parameters, it can set aside a frame for the picture which it will load later, and immediately place all the text on the page.

Unfortunately, inserting these parameters into every IMG tag in a Website is terribly tedious and error-prone. First, you have to use some software to determine the width and height of every image you will use. Next, you have to search for every occurrence of an IMG tag in the site. Then, you have to manually type in the appropriate WIDTH=123 HEIGHT=123. Not only does this process take a very long time, it is irritating. Also, typos during this type of mundane, repetitive work are common, resulting in distorted images and more work.

HTML Image Scanner solves this problem once and for all. With the click of a button it will scan every IMG tag and every referenced image in an entire Website and automatically insert the correct WIDTH and HEIGHT parameters. A number of customizable parameters let you decide, for example, whether to alter an existing parameter (that may intentionally be different from the actual image size) or to leave it alone. You can even enter a list of image filenames which you want HTML Image Scanner to ignore.

Another important attribute in the IMG tag is ALT. This attribute specifies text to be displayed in place of the picture in cases where the picture has not yet been loaded, when the browser is unable to display pictures, when the user has selected not to display pictures, and when Web pages are accessed by the blind. HTML Image Scanner alerts you to every missing ALT attribute, and lets you insert it on-the-fly, complete with the ability to show you the picture then and there.

If you have many images spread across many pages, or if you frequently modify the images included in your pages or add new ones, HTML Image Scanner is an absolute necessity.

HTML to Text Converter

Web authors work with HTML. Even if your source files came from another source, once they have been marked up using HTML, they are no longer viewable without an HTML browser. However, it is often necessary to convert an HTML document back to plain text.

A simple approach would be to simply remove all the HTML markup from a file. This would leave a rather ugly, unformatted text file containing many extra spaces, tabs, and line breaks. A better approach would be to interpret the HTML tags contained in the document, much as an HTML browser does, and create a text file containing some of the formatting from the original. HTML PowerTools' HTML to Text Converter takes this improved approach.

The formatting available in a text file is limited, but HTML to Text Converter maintains many

of the important aspects such as headings, titles, bulleted/numbered lists (even multi-level numbered lists), paragraph breaks (differentiating between
 and <P> tags), horizontal rules (<HR> tags), and more. Of course, all spaces, tabs, and line breaks used to format the HTML source are dealt with intelligently to eliminate extraneous spaces and line breaks in the converted text file.

Unlike other approaches for converting files, HTML to Text Converter saves time and effort by running in batch mode, converting some or all of the files in a Web project with one button click.

Customizable options allow some control of the how the text file is rendered, including choosing word-wrap, setting maximum line length, and choosing to render horizontal rules (<HR> tags) or not. If you ever need to convert HTML documents to text files, HTML PowerTools' HTML to Text Converter will be a very valuable utility in your HTML developer's toolkit.

HTML Date Stamper

Not only is it customary on the Web to include a "last modified on" date in Websites, it is an important indicator to those viewing your pages that the pages are recent and up-to-date. One thing sure to convince a browser not to return to your pages is if they are not updated frequently.

The amount of time and effort required to go into your home page and modify the date is not extreme. However, wouldn't it be beneficial to include a "this site last modified on" date stamp on the bottom of every page in your site? With the Web's new and more powerful search engines (Digital's Alta Vista is a prime example), more and more users will be entering your sites at individual pages and not through your home page. Thus, if you want to show your audience that the site is up-to-date (even if that individual page has not been recently changed), you should have a date stamp on each and every page.

The amount of time and effort to insert the current date into every page, though, is not negligible. HTML PowerTools' HTML Date Stamper will do this for you automatically, on some or all of your Website's pages, with the click of a button.

You provide a simple set of rules that tells the program where to insert the current (or some other) date in any files that you want to be affected. For example, you could define a rule as, "replace all text between the words 'Last modified: ' and the next period with today's date." A number of date and date/time formats are available to choose from.

Visit our Website at <http://www.tali.com>

Project Window

The Project Window, which initially displays all files included the current project, is displayed when clicking Refresh File List from the main window. This window is also displayed automatically when clicking Analyze HTML if the file list has not yet been refreshed. (Note: you can skip this window when clicking Analyze HTML by holding down Shift or Ctrl when clicking that button.)

(Note that if Only scan linked files is selected in the [project definition](#) then instead of a file list you will see only the name of the specified home page from which links will be followed to other HTML files.)

Hint: If you can't see the ends of your file names (because the paths are long), you can drag the window's border to widen it.

If you are satisfied with the file list after refreshing it, simply click OK (to return to the main window, if you clicked Refresh File List to get here) or Proceed (to begin the analysis, if you clicked Analyze HTML).

Alternatively, you have three other options:

- The Modify Project button takes you straight to the [Project Properties](#) window to make changes to the current project's definition. When returning to this window, the file list will be refreshed based on your changes to the definition.
- The Exclude Files button allows you to temporarily exclude certain files from an analysis. After clicking this button, the Project window will contain two lists: one showing all files to be included in the analysis, and one showing all those excluded. Use the Exclude and Include buttons to move a single file from one list to the other, and use the Exclude All and Include All buttons to rapidly move all files from one list to the other.

Hint: double-clicking a file name will move it to the other list.

- The Other button provides a dialog box for selecting a single file to be analyzed. This file may be part of a defined project, but it does not have to be. Note that this has no impact on the project definition, and is a one-time selection. If you do not click Refresh File List, however, you can re-run the analysis on the selected file as many times as you wish without having to reselect it.

Main Window

From the main window of HTML PowerAnalyzer you can navigate to the [Project Manager](#) and [Options](#) windows by clicking the appropriate buttons. You can also begin an [analysis](#) by clicking the Analyze HTML button in the center of the window.

The name of the currently selected project is displayed at the top of the window, as well as the number of files currently selected for analysis. If you used the [Project window](#) to temporarily exclude any files from the analysis, the number of files displayed will indicate this fact.

Clicking Refresh File List will re-scan your hard disk to find all files included the project's definition. The resulting list is displayed in the Project window. There, you can temporarily exclude certain files from the analysis, jump directly to the [Project Properties window](#), or select a single file to be analyzed instead of the entire project.

Note that it is not necessary to click Refresh File List to perform an analysis. Clicking Analyze HTML will automatically cause a refresh and provide you with the opportunity to exclude files or select a single file not in the current project.

Hint: if you do not wish to see the Project window before performing an analysis, you can hold down Shift or Ctrl when clicking Analyze HTML to automatically proceed straight to the analysis itself.

Project Manager Window

This window, accessible by clicking Select Project from the main window, shows a list of all currently defined Website projects. To select one for analysis, click its name and click OK.

To add a new Website project to the list, click New. The [Project Properties](#) window will appear.

To modify an existing project's definition, select it and click Properties. The [Project Properties](#) window will appear.

To remove a project from the list, select it and click Remove.

Note that the Cancel button in this window will not undo changes made in the Project Properties window.

Project Properties

The Project Properties window appears after clicking New or Properties in the [Project Manager](#) window.

After entering all the required details, click OK. To ignore all changes made (and to cancel the addition of a new project), click Cancel.

- Project Name - Enter a descriptive name for the project up to 20 characters in length. You will use this name to identify the project. It will also appear in reports.
- Project Code - Enter a code for the project, up to five characters in length. This code is used as a prefix for report files generated during analysis. If, for example, the project's name was Demo Website, you could use a code prefix of DEMO. The analysis report filename would be DEMO.TXT or DEMO.HTM (depending on the option selected).
- Home Page - Enter the file name of the project's home page (it's full path is not required). If Only scan linked files is selected then this will be the first file analyzed, and it must exist in the directory specified for this project. All HTML files linked to from this home page, and all files linked to files linked to from this home page, etc. will be analyzed. If Only scan linked files is not selected, then the home page entry has no relevance.
- Only scan linked files - When selected, the analysis will begin with the home page file specified, and continue with HTML files linked to from the home page. When not selected, the analysis will scan all the files in the directory (or directories) specified that match the Files to Include entry.
- Directory - Enter the full path of the project's root directory. Use the Browse button to browse your hard disk for the correct directory.
- Include Subdirectories - If the project's HTML files occupy subdirectories below the specified directory, make sure this check box is checked. When Only scan linked files is selected, the only relevance of this selection is in determining the unused (unreferenced) file list: any files in any of the subdirectories that are not explicitly referenced by the project's scanned pages will be listed as unreferenced.
- HTML File Mask - Here, specify one or more DOS-style file masks for your files. Generally, this will be simply *.htm. You may specify multiple file masks by separating them with semicolons, e.g. *.htm;*.shtml. When Only scan linked files is selected, this setting is used to determine which links to follow for the purpose of analyzing a file.
- Refresh - Click this button to ensure that your other entries specify the files that you expect. This button is located here for your convenience only, and its use is not required.
- Files in Project - Here, you can review the files included in your project definition. After changing any of the entries in the Project Location frame, the list will be cleared. Click Refresh to re-scan the indicated files.

Analysis Window

This window is displayed while an analysis is in progress.

- The Progress frame indicates which file is currently being analyzed, as well as the progress of the current analysis.
- The Status frame displays certain key information about the current file and the entire Website being analyzed.
- Always Pause Between Files - If this check box is checked, the analysis will pause after every file analyzed. This gives you the opportunity to click the View File Results button for each file (see below). If the Always Pause Between Files option is selected in the [Options](#) window, then this check box will be checked at the start of every analysis. Otherwise it will be unchecked. You can change its status by clicking it at any time. Note that if either of the two Pause After Files options are selected in the Options window, the analysis may pause between files even when this check box is not checked here.
- View File Results - You can click the View File Results button to see the [File Results window](#), which provides a "real-time" report on the **current** file. This button is only available when in pause mode, either because the program paused due to selected [options](#), or because you clicked the Pause button.
- Launch Report Viewer - The View File Results button becomes Launch Report Viewer at the conclusion of an analysis. If Windows has an application associated with the report file type (HTML or text), that program will be launched and the current report will be loaded.
- Pause - Click this button at any time to pause the analysis in progress. When in pause mode, this button will become Continue, used for continuing an analysis (ending pause mode). When in pause mode, you can click View File Results to see a report on the current file, or Close to abort the analysis.
- Close - If you click this button in the middle of an analysis, no report will be written to disk. After an analysis, Close will simply close this window. Close is only available in pause mode (see above).

File Results Window

The File Results window is accessible from the analysis window during an analysis. If viewing this window in the middle of a file's analysis, the information displayed may not yet be complete.

Note: this window only shows results for the current file, not the entire project being analyzed.

This window contains seven lists. The first contains all HTML errors found in the file. The other six, selectable by clicking one of the option buttons on the left of the window, contain alphabetical lists of links as indicated. To the left of the option buttons are the number of links (a.k.a. references) included in each category. Categories with no links are disabled. Note that links will only be analyzed if Check internal links is selected in the [Options](#) window.

- All - a list of all links found, as they appear in the original HTML file.
- External - a list of all external (a.k.a. remote) links. These will all begin with a protocol indicator, such as http://, ftp://, and mailto:
- Internal, found - a list of all internal (a.k.a. local) links that have been verified. If the reference contains an anchor target (e.g., index.htm#intro), the existence of the anchor has been verified. The reference, as it appears in the original HTML file is listed, with the interpreted absolute path in parentheses.
- Internal, missing - a list of all internal (a.k.a. local) links that HTML PowerAnalyzer has not been able to verify for some reason. Most likely, the referenced file does not exist, or an error occurred while attempting to read it. If the reference contains an anchor target (e.g., index.htm#intro) whose file exists but whose anchor target does not exist, this will be mentioned (if there is not enough room in the view window to see this indication, you will have to refer to the report at the conclusion of the analysis). The reference, as it appears in the original HTML file is listed, with the interpreted absolute path in parentheses.
- Unrecognized - a list of all links which HTML PowerAnalyzer was unable to resolve. Unknown protocols (e.g., ticker://abc.com) and incorrect syntax (e.g., dir//file) are the most common reasons for an item to appear in this list. Note that using the [HTML Rulebase Editor](#) you may define additional protocols for recognition during analysis.
- With capital letters - a list of all references containing at least one capital letter. This can be particularly useful for locating links that will function correctly in testing in the case-insensitive Windows/DOS environment, but will fail in the case-sensitive UNIX environment. For example, when testing a link on your PC, INDEX.HTM, Index.htm, and index.HTM will all work. However, if the file index.htm is on a UNIX Web server, the hyperlink will fail.

HTML Rulebase

An HTML Rulebase is a file containing all relevant facts about the HTML markup language, and is the logical basis for all of HTML PowerAnalyzer's decisions. Rulebase files have an HRB extension and are located in the directory where the HTML PowerTools programs reside.

The contents of an HTML Rulebase can be customized using the [HTML Rulebase Editor](#), available separately. Up-to-date Rulebase files can also be downloaded from www.tali.com.

About HTML PowerAnalyzer

HTML PowerAnalyzer is one of Talicom's HTML PowerTools. To see which version you are using, right-click on the main window's Help button.

Please refer to <http://www.tali.com> for complete information about Talicom's line of HTML PowerTools for Windows.

Silent Mode

HTML PowerAnalyzer can run Silent Mode, during which no user interaction is required to analyze a single HTML file or an entire Website project. The program will start up, perform the analysis, generate a report, and shut down all by itself.

You can run HTML PowerAnalyzer in Silent Mode to analyze an entire Website project by defining a command line for the program's icon. Simply add the code of the project that you wish to analyze. The following is an example:

```
c:\program files\html powertools\analyze.exe PROJ1
```

You can also run the program in Silent Mode to analyze a single HTML file. This is useful if you have just completed modifications on a single file and you have no need to re-scan an entire Website project. There are two ways to do this:

1. Define a command line for the program's icon with the HTML filename as a parameter. For example:

```
c:\program files\html powertools\analyze.exe d:\website\index.html
```

2. Drag and drop an HTML file from Windows File Manager or Explorer onto HTML PowerAnalyzer (either onto its name in File Manager or Explorer, or onto its icon).

Note: if the analysis is interrupted, either by user interaction (such as clicking Pause) or by an error, the program will not automatically close.

Customize Report Template Window

This window, accessible from the Report tab of the [Options](#) dialog, is used to modify an existing report template, or to create a new one.

If you leave the Name entry unchanged, the current template will be modified and saved to disk when you click OK. If you change the name, a new template will be created with that name and saved to disk when you click OK. It is recommended to always change the name when altering one of the supplied templates so that you can return to it in the future if so desired.

In addition to general template properties, each report template contains two distinct sections: [Project-level](#) and [File-level](#). The former appears once, at the beginning of each report, and includes information on the project as a whole. It also may include (in HTML format reports) links to individual HTML files included in the report. The latter is repeated in the report, once for each HTML file included. You can edit each section by selecting the appropriate tab and making changes to the textual content of the report.

In addition to "literal text" that will be included in the final report, there are a number of [field codes](#) that are used to represent information generated by the report itself. Field codes are represented by a name surrounded by curly brackets { }. To easily insert a desired field code, simply select it from the drop-down list box below the text editor section and click Insert. See the list of available field codes, below. Only one field code may appear on each line. Note that some field codes represent a number or some text while others represent a list of items.

The filename of the report template, as stored on disk, is displayed in the title bar of the window. The filename is displayed for informational purposes only; you do not need to know it to use it in the program.

Template Properties

- HTML format / Text format - If HTML tags are included in the report template, select HTML format. If you wish to generate a simple text file report, select Text format. This selection has major implications for the generation of the report during an analysis: if HTML format is selected, items in the report are generated with HTML tags that provide cross-indexing. In Text format reports, these items are reported as simple text.
- Report only files containing errors - When selected, the report will only contain information on individual files that contained errors or broken links. This provides the advantage of a shorter, easier to view report. When not selected, individual file information for all HTML files analyzed will be included.
- Report only files containing errors - When selected, the report will only contain information on individual files that contained errors or broken links. This provides the advantage of a shorter, easier to view report. When not selected, individual file information for all HTML files analyzed will be included.

Project-level Report Template

As mentioned above, this section includes information pertinent to an entire Website project, and to the analysis as a whole. You can edit the text itself in the edit window. You can insert a field code by selecting one from the list and clicking the Insert button.

File-level Report Template

As mentioned above, this section includes information pertinent to individual HTML files analyzed. You can edit the text itself in the edit window. You can insert a field code by selecting one from the list and clicking the Insert button.

Field Codes

The following field codes are available for the Project-level Report Template:

{AnalysisTime}	Date and time of analysis
{ListAllFiles}	List of all files in project
{ListBadLinksFiles}	List of all project files with invalid links
{ListErrorFiles}	List of all project files with HTML errors
{ListExternalLinks}	List of all external links in project
{ListUnlinkedFiles}	List of all unreferenced files in project directories
{NumFilesAnalyzed}	Number of files analyzed
{NumHTMLErrors}	Number of HTML errors in project
{ProgramVersion}	Program version number
{ProjectName}	Name of the current project
{ReportFileName}	File name of the current report
{RulebaseUsed}	HTML Rulebase used for analysis
{TotalBytes}	Total byte count of all files analyzed
{TotalLines}	Total line count of all files analyzed

The following field codes are available for the File-level Report Template:

{HTMLFileName}	HTML file name
{HTMLFileTime}	HTML file's date and time
{ListAllExternalLinks}	List of all external links in file
{ListHTMLErrors}	List of HTML errors in file
{ListHTMLTags}	List of HTML tags used in file
{ListInvalidInternalLinks}	List of invalid internal links in file
{ListLinksWithCaps}	List of links containing capital letters
{ListUnknownLinks}	List of unrecognized links
{ListValidInternalLinks}	List of valid internal links in file
{NumAllExternalLinks}	Number of all external links in file
{NumInvalidInternalLinks}	Number of invalid internal links in file
{NumLinksWithCaps}	Number of links containing capital letters
{NumUnknownLinks}	Number of unrecognized links
{NumValidInternalLinks}	Number of valid internal links in file
{TotalBytes}	Number of bytes in the current file
{TotalLines}	Number of lines in the current file

