

yesyesyesWebWatch Specter Inc.  
1995yesyesyesWebWatchTRUEwebwatchyesyes02/04/95

## Table of Contents

Purpose and Usage  
Table of Contents  
Main Dialog  
Progress Dialog  
Future Plans  
Feedback  
Unattended Run  
webwatch.ini  
Future Plans

Help file produced by **HELLLP!** v2.5 , a product of Guy Software, on 4/2/95 for Unregistered User.

The above table of contents will be automatically completed and will also provide an excellent cross-reference for context strings and topic titles. You may leave it as your main table of contents for your help file, or you may create your own and cause it to be displayed instead by using the I button on the toolbar. This page will not be displayed as a topic. It is given a context string of \_\_ and a HelpContextID property of 32517, but these are not presented for jump selection.

HINT: If you do not wish some of your topics to appear in the table of contents as displayed to your users (you may want them ONLY as PopUps), move the lines with their titles and contexts to below this point. If you do this remember to move the whole line, not part. As an alternative, you may wish to set up your own table of contents, see Help under The Structure of a Help File.

Do not delete any codes in the area above the Table of Contents title, they are used internally by HELLLP!

## Table of Contents

Purpose and Usage  
Unattended Run  
Main Dialog  
Progress Dialog  
webwatch.ini  
Future Plans  
Feedback

## Purpose and Usage

WebWatch is an Internet utility that you can use to track changes in selected Web documents.

As **input** to the program you specify

- ☐ an HTML document, referencing the URLs you want to track, and
- ☐ the date after which you consider the document to be updated. (E.g. the date when you last ran WebWatch.)

WebWatch will **generate** a local HTML document, containing links only to those documents which were updated after this date. You can use this local file to navigate to the updated documents, using any Web browser.

WebWatch stores its arguments in a webwatch.ini file. This way, once you customize the program to your needs you can run it with a single mouse click or even unattended. All arguments are set in the main dialog. In addition to setting these arguments in the dialog, the program automatically updates the value of the *last modified date* field. After a complete run it is set to the **beginning** of the run, so that you can be sure that you never miss an update of one of the specified URLs; but at the same time eliminating redundant retrievals. Of course if for some reason you want to change this field (e.g. you didnt have time to browse all the updated URLs and want to refresh your catalog again from an earlier date, or you are in a hurry and want to see only last minute changes and want to set the date to a more recent one), you can do that too.

If you become a regular WebWatch user, you may set the *anchor document* (the local file, parsed for your URLs) to your Netscape *bookmark.htm* file or to a similar collection of URLs and the result file (the file where the program generates its output) to your *home page*. This way your home page will always contain the fresh, must see documents; and you still can use your bookmarks in the usual way.

## Unattended Run

Once you become comfortable with the operation of WebWatch you may rely more and more on its results.

To query a large number of documents for their last modification date may take a long time, especially during rush hours. For this reason, WebWatch offers an *unattended mode*. You can invoke the program from the command line (or from a script) with the -u flag, i.e. *WebWatch -u*. The program will read its arguments from the webwatch.ini file and runs to completion. The progress dialog is visible in this mode too, so you can cancel or abort the operation.

## Main Dialog

When WebWatch starts, you can define the parameters for its operation. Initial values of these fields are taken from the webwatch.ini file.

The meaning of the individual controls:

**Anchor Document** and **Browse...** The document identified by these controls (Browse simply fills the edit control using the standard File Open dialog) contains the URLs to be visited. The document is fully parsed for embedded URLs, everything else is quietly ignored. For example a *bookmark.htm* file as maintained by Netscape can be used, but any other HTML document, even if it contains graphics, free text, etc. will do. You may have created this document yourself, or have it generated by a program or even downloaded it from the Internet. Text, graphics, formatting information, generally anything, but URLs are not copied into the result file.

The value of this field is stored in the *LocalAnchor* field of the webwatch.ini file.

**Include URLs newer than.** Only URLs newer than this date will be included in the *Result File*. The format of this entry is rather flexible, most reasonable date/time formats are accepted. The value of this field is converted to a standard format and written into the *Date* field of the webwatch.ini file. [But you can edit this value in the .ini file even after the conversion, using any other format.] After a successful run the program updates this value to the beginning of the run.

**Include URL-s with unknown date.** If this checkbox is cleared, only URLs, which return a **definite** *Last Modified* date from the server **and** this date is newer than the one specified in the *Include URLs newer than* field will be included in the Result File. If this checkbox is set, than **in addition** to these URLs those with unknown dates will be included in the result too.

The date of a URL can be unknown for a number of reasons:

- ☐ the request may time out [see *Time out after (sec)* field];
- ☐ the request is successful, but the server doesn't return the Last Modified date of the document. This is an optional field in the HTTP protocol and servers may decide not to use it;
- ☐ a client or server error is detected. These errors (4xx and 5xx error codes) will be included in the result file. We at Specter Inc. can't do much about server errors but if you let us know about client errors, we will make every effort to fix them. Note that *Document Moved* return codes (3xx) are not considered errors. The program will follow the link and will record both your original and the updated URL in the result file. (You may use this information to update your original anchor document. This way next time the program can access the document with a single request.)

The value of this field is included in the *NoDate* field of the webwatch.ini file as integer. (1 indicates checked, 0 cleared state.)

**Result File:** and **Browse...** The result will be placed into this file. The result is an HTML document (with .htm extension), which can be loaded using the appropriate Local File command from the File menu of any Web browser. The *Browse...* control fills the field with the selected file and prompts for overwrite confirmation if you select an existing file. In unattended mode this prompt is suppressed; you have to be sure that if you specify an existing file name here, you don't mind that the file will be overwritten.

The value of this field is stored in the *Result* field of the webwatch.ini file.

**Time out after (sec):** in this field you can enter an integer after which the program should abandon the **current** retrieval and continue with the next. This is **not** an aggregate value; i.e. if you specify 10 here and the *Anchor Document* references 100 URLs, the program **may** run for 1,000 seconds. This field is advisory only. The program will make every attempt to cancel the current retrieval after this period, but there are no guarantees that it can do so. When a retrieval can be actually interrupted, depends on the used TCP/IP stack.

Setting this field to 0 (zero) indicates that the program shouldn't try to time out. This does **not** mean that once the program tries to contact a non-responding server it will hang forever. The TCP/IP stack will time out, but the strategy (value) for this is implementation dependent.

The value of this field is stored, as integer, in the *TimeOut* field of the webwatch.ini file.

**Run...** Checks the validity of the fields and if everything is correct, starts the retrieval. During retrieval the program's progress can be monitored (and interrupted) by the progress dialog. While that dialog is displayed, the controls of the main dialog are blocked (can not be edited.) After completing the full retrieval the progress dialog is dismissed and control returns to the main dialog.

The unattended mode in effect pushes this button automatically. This means, that if a field (as taken from the webwatch.ini file) is not correct, the retrieval will not start; it will display the error message box and will wait for somebody to dismiss it, fix the problem and press the *Run* button again.

**Help** Displays help. (This file.)

**Done.** The program is dismissed.

## Progress Dialog

While the program is running this dialog is displayed. After all documents have been retrieved (or the user has aborted the run), control returns to the [main dialog](#).

This dialog shows the **Name** and the **URL** of the document being retrieved (as stored in the *Anchor Document*.)

A group of 3 check boxes indicates the **progress** of the retrieval:

- ☐ **Host Contacted** is checked once the connection to the host has been established.
- ☐ **Request Sent** is checked when the request to retrieve the headed of the URL has been posted. Generally this comes very quickly after the connection has been established.
- ☐ **Response Received** is checked when the header from the server is retrieved. Generally you wont even notice this checked because processing the header and generating the output is very fast and the program immediately proceeds to the next URL (and the check boxes get cleared.)

With the **Cancel** button you can cancel the current retrieval. Program execution continues with the next URL.

Pressing the **Abort** button aborts the current run. Control returns to the [main dialog](#).

Note that canceling or aborting retrieval does not interfere with the generation of the output file. The output is built as we go, so even after abort the result is available.



## webwatch.ini

The webwatch.ini file stores all parameters necessary for a run or an unattended run of the program. It should be placed into the Windows directory. It can be edited using any ASCII text editor. Its fields and their interpretation (for more detailed information please refer to the description of the main dialog) are:

- ☐ **Date** Only documents with modification date later than this will be stored in the result file.
- ☐ **Result** Points to a valid, fully qualified HTML file. (If the file doesn't exist, it will be created.) The results of the run will be generated into this file.
- ☐ **LocalAnchor** Points to an existing HTML file. URLs referenced in this file will be visited and checked for last modification date.
- ☐ **TimeOut** Integer. Indicates (in seconds) how long the program should try establish connection to a server.
- ☐ **NoDate** Integer, possible values are 1 or 0. If set (1), documents whose last modification date can not be determined will be included in the result.

## Future Plans

The base technology (parsing HTML documents outside of a Web Browser) will be used extensively in future Specter products, not necessarily related to WebWatch.

WebWatch itself can be extended in a number of ways (and the feedback we receive during the beta cycle may influence our plans!):

- ☐ The last modified date can be specified for individual URLs separately.
- ☐ Documents, recognized as fresh (or some of them, see filtering below) can be loaded into the cache of the browser. All the leg work can be done in batch via WebWatch and the user can use his time for reading the documents.
- ☐ In addition to the date other criteria for inclusion in the result set can be specified; e.g. Did my favorite author contribute? or Was my favorite topic mentioned? This (client side filtering) is the topic of some new experiments and research.
- ☐ Other protocols (gopher, ftp, news) will be added.
- ☐ The anchor document can be taken from a remote site instead of a local file. This will accomodate users who for their browsing wish to rely on the experience and recommendations of a professional content-provider, editor or guru.
- ☐ WebWatch can go recursive, i.e. can visit not only URLs mentioned in the anchor document but URLs mentioned in referenced documents, and so on. This can easily lead to traffic explosion. With a few mouse clicks and referencing a relatively short top-level document the user can start a retrieval, traversing the entire Internet, many times over. How to control this (without giving up the usefulness of WebWatch) is the topic of another set of experiments and research.

## Feedback

We appreciate that as a beta tester you take the time providing us with feedback. Please send any comments to

**E-mail:** [webwatch@specter.com](mailto:webwatch@specter.com)

**US Mail:**  
Specter Inc.,  
45 Selwyn Rd.  
Belmont,  
MA 02178

**Fax:** (617)484-6512

Please visit our **Web page** for information about new products and releases: <http://www.specter.com/users/janos/webwatch/>

If you send us bug reports, please try to provide as much information as you can to help us fix the problem:

- ☐ Where did you get the software, which version are you using. (The About box can be invoked from the System Menu of the Main dialog);
  - ☐ What is the platform, configuration. TCP/IP stack, memory, speed and type of your Internet connection;
  - ☐ The document you tried to retrieve (its full URL..)
  - ☐ Circumstances how the retrieval failed or the program crashed. E.g.: size of anchor document? Was this URL at the beginning of it or more towards to the end? Could you isolate the problem (reproduce without the rest of the anchor document)?
- Were you running other winsock clients when the problem occurred?

Every bug report, with or without full details and any other comments will be gratefully acknowledged.



