

yesyesyesWebWatch Specter Inc.  
1995yesyesyesWebWatchTRUEwebwatchyesyes12/08/95

## Table of Contents

[Purpose and Usage](#)

[Table of Contents](#)

[Main Dialog](#)

[Progress Dialog](#)

[Future Plans](#)

[Feedback](#)

[Unattended Run](#)

[webwatch.ini](#)

[Future Plans](#)

[Proxy](#)

[system menu](#)

[Result file](#)

[Registering on the Web](#)

[Registering through CompuServe](#)

[Credit Card Orders](#)

[Ordering from Specter](#)

[UCT](#)

[Last Visit](#)

[License Agreement](#)

[Authentication](#)

Help file produced by **HELLLP!** v2.5 , a product of Guy Software, on 8/12/95 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

Proxy

Authentication

Unattended Run

Main Dialog

Progress Indicators

Webwatch.ini

Result File

Technical Support

Registration

License Agreement

## Purpose and Usage

WebWatch is a tool for keeping track of changes in selected Web documents.

As **input** to the program you specify

- an HTML document that references the URLs you want to **track**, and
- the **date** and time after which you consider updates to the referenced documents to be interesting. You also have the option to override this date with the time of your **last visit** to each referenced document, as recorded by your Web browser.

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

WebWatch **saves time** several ways:

- you don't need to check every URL you monitor just to learn that nothing has changed. "We watch the Web so that you don't have to." WebWatch's output indicates for you the documents that have changed and need your personal attention;
- WebWatch takes its input from a file. In a single run you can check all the documents you are currently tracking. There is no need to sit in front of your browser clicking and checking each document separately;
- WebWatch retrieves only the "header" of a document, to check its "Last Modified" date. The size of this header is usually quite small compared to that of the entire document. With WebWatch, your connect time, and ultimately the load on the network, is significantly reduced.

If you are a **content provider**, you should encourage your subscribers to use WebWatch. If they put your server on their list, they will be alerted by WebWatch when you change your content. You can be "dormant" for a while, preparing the "big one," without worrying that you drop off the beaten path. WebWatch, unlike humans will loyally and dutifully check the date on your documents, day in, day out. Or you can publish time sensitive data without worrying that "once a week" users will not see it for six days. Also, the load on your server and network connection will be reduced, since WebWatch fetches only the header of each document.

If you are a **connectivity provider**, your customers will appreciate if you point out to them this productivity improvement tool. The load on your server will be less, you can accomodate more customers.

WebWatch stores its parameters in the webwatch.ini file. Once you have customized the program, using the controls in the main dialog, you can run it with a single mouse click or even unattended, periodically. In addition, the program automatically updates the value of the "Date" parameter. After a complete run, it sets this parameter to the time the run *started*, so that you won't miss updates of any of the specified URLs (even if they are updated during the WebWatch run.) Of course, if you want to change this parameter, you can do that too. You might want to change manually the "Date" parameter if, for example, you didn't have time to browse all the updated URLs, and want to refresh your catalog again from an *earlier date*; or if you are in a hurry, and want to see only "last minute" changes and want to set the date to a *more recent* one)

If you become a regular WebWatch user, you may want to set the *"anchor file"* (the WebWatch input file containing URL references) to your Netscape **bookmark.htm** file, an exported Mosaic **hotlist**, or a similar collection of URLs; and set the "result file" (the file where WebWatch places its output) to your *home page*. Your home page will always contain the fresh, "must see" documents; and you can still use your bookmarks in the usual way.

## Proxy

The Proxy... dialog can be invoked from the system menu. A proxy server can be identified either by its IP address, like 204.156.22.13 or by its name, like wwwcache.hensa.ac.uk. Do **not** use qualified names, like http://wwwcache...

The port of the server must be specified.

You can maintain a **no proxy** list. This is a list of servers which serve to you documents **directly** without going through the proxy server. This list has no effect if proxy server is not specified.

To **add** a server to this list, type the domain name or IP address of the server into the provided edit box and press the **Add** button. The name will disappear from the edit box and is added to the end of the list.

To **delete** a server from this list, highlight its name in the list box and press the **Delete** button.

## Authentication

Authentication is a process when a server or service verifies whether you are entitled to use it. It requires entering a valid {userid, password} pair. Once this data is entered, WebWatch encodes it and submits it on your behalf to the requesting server. If the server accepts the data, program execution continues. If the authentication fails (e.g. you mistyped your password), WebWatch prompts you again. If you wish to ignore the URL requesting authentication (e.g. you forgot your password), click on Cancel. WebWatch will continue with the next URL in the anchor file.

The {userid, password} pair is stored in the webwatch.ini file (encoded.) Next time you run WebWatch (or during the current run if a later URL references an already authorized server), WebWatch will not prompt you for this data; it will read the data from the .ini file. This is especially useful if once you set up your userids, you wish to run the program in unattended mode.

Two kinds of authentication are supported:

**Proxy-authentication** is required by the proxy server. This you give only **once** as WebWatch cant change proxy servers in the middle of a single run. Note that WebWatch will not send this userid to any no-proxy servers on your list.

**User-authentication** is required by certain servers and services. You may have **arbitrary** number of such URLs in your anchor file. WebWatch will prompt you for your userid on each *server*. You may query several URLs from the same server: WebWatch will prompt you only once.

## Unattended Run

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

To query a large number of documents for their last modification dates 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 `dialog` is visible in this mode too, so you can cancel or abort the operation.

The program handles authentication requests first checking the `webwatch` file for a valid (encoded) {userId,password} pair, and if this fails, prompting the user for this data. In unattended mode this would result in a dialog waiting for user input, while the user may not be in front of the screen. To avoid this problem, in unattended mode WebWatch skips URL where authorization failed (because no userId was given or it was invalid.) These documents will be marked with the appropriate Authorization failed error message in the output file.

## 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.

**Anchor File** 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 or a *hotlist* exported by Mosaic 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 a Web site. Text, graphics, formatting information, generally everything but URLs is ignored.

The value of this field is stored in the *LocalAnchor* entry of the [webwatch.ini](#) file.

**Include URLs newer:** 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 time the run started.

The value of this field is stored in the *Date* entry of the [webwatch.ini](#) file.

**Use Last Visit date instead (if available)** If this checkbox is set, WebWatch checks, for each URL in the anchor file, whether a "Last Visit" date is stored. If it is stored and is non-zero, this date will be used instead of the date given in the *Include URLs newer* field.

If this checkbox is cleared, the time specified in the "*Include URLs newer*" field is used unconditionally.

The value of this parameter is stored in the *Last Visit* entry of the [webwatch.ini](#) file as integer. (1 indicates checked, 0 cleared state.)

**With unknown date.** If this checkbox is cleared, the only URLs included in the result file are those with headers containing a *definite* "Last Modified" date that is later than the date specified in the *Include URLs newer* field (or the "Last Visit" date, if applicable). If the checkbox is set, then URLs with unknown dates are included in the result file as well.

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

- the request may time out [see *Skip after* 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. Then, the next time, WebWatch can access the document with a single request.)

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

**Skip after (seconds):** this field specifies a number of seconds 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 File* 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 in every case. The actual points during retrieval when it can be **safely** interrupted, depends on the platform and on the installed TCP/IP stack.

The value of this field is stored, as integer, in the *TimeOut* entry of the [webwatch.ini](#) file.

**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.

**Start.** 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. During retrieval the fields that define parameters for the program's operation are blocked and can not be edited. After the retrieval has completed, the fields can be edited again.

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

**Skip** This button is enabled only during retrieval. Clicking this button results in canceling the retrieval of the **current** URL at the *earliest possible time* when this can be done safely. This depends on the state of the open communication channel.

**Stop.** The same button which **starts** the retrieval is used to **stop** it (the label on the button changes.) Pressing this button terminates the retrieval at the earliest time when it can be safely done. [This depends on the state of the open communication channel.] If the you don't push this button during retrieval, the program will *revert* into "editing mode," (when modifying parameters is enabled) automatically after each document has been checked.

Skipping or stopping retrieval does not interfere with the generation of the output file. The output is produced incrementally, so even after "stop" the result (up to that point) is available and saved.

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

**Done.** The program is dismissed. This command is disabled during retrieval.

## Progress Indicators

When the program is **retrieving**, a set of controls, at the bottom of the main dialog shows the progress of each retrieval as well as the total progress.

The **Name** and the **URL** of the document being retrieved are shown.

A **progress indicator** shows the percentage of documents retrieved. This is **not** the same as total time. This latter can not be calculated in advance, as response time varies from document to document.

Under the progress indicator two labels show the progress of the retrieval. The label indicating current activity (what the program is waiting for) is highlighted:

- Connecting to host...**
- Waiting for response...**

## Result file

The generated result file consists of three parts:

The **header** contains a tabulated description of the parameters used to generate the file; e.g. the input file name and the setting of various variables;

The **list** has one entry for every URL which was **requested** to be displayed. URLs that were filtered out because of the parameter settings are omitted from this list. For "*Moved Documents*" (documents which are not at the address you used in the anchor file, **but** which left a "forwarding address") an entry is generated, *regardless* of whether the target document itself was filtered out. If the [moved] target document was not filtered out, the next entry references it. In other words, if you see an entry indicating that "Document moved", but the very next entry is **not** the URL of the document in its new location, you know that the document was filtered out.

An **entry** in the list contains two or three fields. The URL of the document is always displayed (as a hyperlink), followed by the Last Modified date, as retrieved from the server. The third field is present or not depending on the setting of the Last Visit override parameter. If that parameter was set, the third field displays the date of your last visit to the document (even if it is not available, indicated by N/A.)

Authentication errors are listed only once and only if the user cancelled the userID-prompt. I.e. if the user mistypes the userID several times, but eventually succeeds in getting through the server, the properly authenticated document will show up only once, with the obtained statistics. Note that in unattended mode the program doesn't prompt for userID. This means that every URL which does not have a valid userID stored (during a previous run) in the .ini file will show up with authentication error.

The bottom of the output shows **statistics** (number of documents visited, displayed, total time, etc) about the run.

## 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.
- HTTP\_Proxy\_Server** The name of the proxy server as specified in the *Proxy* dialog
- HTTP\_Proxy\_Port** An integer. The port on the server.
- NoProxy** Comma separated list. Names of servers, accessible directly (without the proxy server.)
- UserName** The name of the registered user as typed in the *Registration* dialog, or UNREGISTERED if this copy of WebWatch is not registered yet.
- RegistrationKey** Registration key and some bookkeeping information.  
Don't edit these two fields by hand. (Use the Registration dialog to change values here.) Other parameters can be edited (but there is no reason to do so; every parameter value is accessible from dialogs.)
- WWW-Authentication** For every server which required {userid,password}, WebWatch stores the encoded key in the .ini file. Next time you run the program, it will not prompt for your userid. This is so that you still can use the program in unattended mode.
- Proxy-Authentication** If the proxy server you are using requires authentication (userid, password), WebWatch will prompt for these the first time you run the program. The values you enter are saved in the .ini file, so next time you don't have to enter them again.

For more information see the help topic on authentication.

## Technical Support

We tried hard to release a thoroughly tested, smooth product. We cant gurantee that the product does not contain bugs. We can gurantee though that we will continue to support it and fix bugs and/or offer work-arounds as quickly as we can.

Registered and unregistered users reporting bugs will receive the same attention. We dont believe that users should be blackmailed into registering just in order to be listened to. As a matter of fact you are encouraged **not** to register (for not more than 30 days) and evaluate the product and test it against your individual needs and configuration. If, during this evaluation period you discover and report a new bug, we will promptly send you a **free registration** as our way of thanking you for your time and apologizing for the inconvenience the bug may have caused you.

Our **Web site** (<http://www.specter.com/users/janos/specter/>) will always contain an up-to-date list of known bugs and also a continuously edited (based on **your** comments) list of *Frequently Asked Questions*.

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 surrounding the retrieval failure or the program crash. 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?

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

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

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

## Registration

After you receive your registration key you must enter it into the *Registration Dialog*. The dialog can be invoked from the system menu.

Single user registration is US \$18. For volume discount, customized versions of the program, source code licensing please contact Specter Inc.

You can use the program for evaluation purposes for 30 days, without registering it. After 30 days you have to register it (even if you are entitled to a free registration) or delete it from all of your systems.

When registering, make sure that you give us a **User name** (to be used to generate the Registration key) and your **e-mail address** (where the registration key will be mailed.) The User Name must be at least 4 characters long, may contain spaces and is case sensitive. If you dont wish your Registration Key sent to your via e-mail, please contact Specter, Inc. for alternative arrangement.

You can register

- through the Web, if you have a First Virtual account (or willing to open one);
- in CompuServe if you have a CompuServe account;
- calling a toll-free number if you want to charge the registration fee to a credit card; or
- directly with Specter, Inc.

## Registering on the Web

If you have a First Virtual account, we can charge the registration fee against it. To order the registration key you can fill out the order form at our Web site at <http://www.specter.com/users/janos/specter/emailpay/html>

If you dont have a First Virtual account, but you are interested in opening one (it costs \$2), you can do so over e-mail in a matter of minutes. Send a (no obligation) e-mail message to [apply@card.com](mailto:apply@card.com) (empty subject field, empty body.)

For more information about First Virtual, visit <http://www.fv.com>

## Registering through CompuServe

If you have a CompuServe account, you can register on CompuServe. To access the registration database, GO SWREG in CompuServe.

WebWatch is **Registration ID 5680** and its price is US \$18.

The Registration Key will be e-mailed to your CompuServe mailbox as soon as CompuServe notifies us about your order (generally the same day.) Your next CompuServer invoice will contain the charge.

## Credit Card Orders

You can order with Master Card, Visa, Amex or Discover from Public (software) Library by calling **800-2424-PsL** (800-2424-775) or 713-524-6394 or by Fax to 713-524-6398 or by CIS E-mail to 71355,470 (from internet: 71355.470@compuserve.com) You can also mail credit card orders to PsL at P.O.Box 35705, Houston, TX 77235-5705.

The product id is **11995**, the price is US \$18. When ordering, make sure to give your User Name and e-mail address, necessary for generating and sending you the registration key.

The above numbers are for credit card orders only. Specter, Inc. can not be reached at these numbers.

Any questions about the status of the shipment of the order, refunds, registration options, product details, technical support, volume discounts, dealer pricing, site licenses, non-credit card orders, etc must be directed to [Specter Inc.](#)

## Ordering from Specter

You can order directly from us either filling out the [First Virtual](#) order form on our Web site or sending cash, check or money order to

Specter, Inc.  
45 Selwyn Rd.  
Belmont  
MA 02178

Single user registration is US \$18. Make sure to include your **User Name** and **e-mail** address, necessary for generating and sending you the registration key.

## License Agreement

You are permitted to use this unregistered shareware version of WebWatch free of charge as a single user on an evaluation basis for a period of up to thirty days. After such period expires, you must either discontinue use of the product and remove all copies from your system, or register the software.

YOU MAY NOT DISTRIBUTE WEBWATCH BY ITSELF OR WITH ANY OTHER PRODUCT, NOR INCLUDE WEBWATCH IN ANY COLLECTION OF SOFTWARE (excluding public anonymous FTP servers and bulletin board systems) FOR PROFIT OR OTHERWISE, WITHOUT THE EXPRESS WRITTEN CONSENT OF SPECTER, INC.

Specter Inc, disclaims all warranties relating to this product, whether express or implied, including but not limited to any implied warranties of merchantability and fitness for a particular purpose, and all such warranties are expressly and specifically disclaimed. The company will make every reasonable attempt to provide bug-free and safe software, but is never under legal obligation to do so. Neither Specter, Inc. nor anyone else who has been involved in the creation, testing, production, delivery, or distribution of this product shall be liable for any indirect, consequential, or incidental damages arising out of the use or inability to use such software, even if the Company has been advised of the possibility of such damages or claims. In no event shall the Companys liability for any damages ever exceed the price paid for the license to use the software, regardless of the form of the claim. The person, organization, or other entity using the software bears all risks as to the quality, performance, use, and misuse of the software.

It is the user's responsibility to determine whether the product will work reliably on their equipment and for their specific needs. The Company does not imply in any manner that this software is suitable for any given application or purpose.

SHOULD YOU DISAGREE WITH THESE TERMS AND CONDITIONS, DELETE THE PRODUCT FROM YOUR COMPUTER(S) AND DESTROY ANY AND ALL OF THE PRODUCT DISTRIBUTION DISKS. THE AGREEMENT IS NOT TERMINATED UNTIL ELIMINATION OF POSSESSED COPIES IS COMPLETE, AND WILL IN NO WAY BE RETROACTIVE. YOUR CONTINUED USE OF THE PRODUCT INDICATES YOUR ACCEPTANCE OF THESE TERMS.

Specter Inc. may be reached via the following address for support questions and general information:

Specter, Inc.  
45 Selwyn Rd.  
Belmont,  
MA 02178  
webwatch@specter.com  
(617)484-6512

The system menu can be dropped down from the system ventillator, the rectangular button at the very left side of the title bar. (Or, on Windows95 clicking anywhere in the title bar with the right mouse button.)

UCT. Universal Coordinated Time: the time elapsed since January 1, 1970, midnight, GMT, in seconds.

The *Last Visit* field was introduced by Netscape Navigator. Netscape maintains this field in its local *bookmark* files. The value is given in UCT format.

