Thunderstone Webinator
WWW Site Indexer Version 24.0.1

Thunderstone Software

April 11, 2022

Contents

Chapter 1

Document Conventions

Webinator runs on Windows Server 2008/2012 and later versions of Windows. This document refers to all versions of Windows simply as Windows.

Webinator runs on many versions of Unix and Unix-like operating systems. This document refers to all variations simply as Unix.

All filesystem and URL paths are based on the default installation location. $INSTALLDIR$ is sometimes used to indicate the directory into which you installed Webinator. The default location for Unix is `/usr/local/morph3`. The default location for Windows is `C:\Program Files\Thunderstone Software\Webinator`.

Examples of command lines and URLs may be broken into multiple lines to fit the printed page. You should not split them when entering them at a command prompt. The split is indicated by $\rightsquigarrow$ at the end of the printed line and $\hookrightarrow$ at the beginning of the next printed line.

```
http://www.example.com/this/a/long/URL/with/many/⇝
   ↪subdirectories/that/will/not/fit/on/a/line.html
```

If a space is required between the two portions, it is indicated with □.

$INSTALLDIR$`/bin/texis profile=`$PROFILENAME$□⇝
   ↪$INSTALLDIR$`/texis/scripts/webinator/dowalk/dispatch.txt`

Chapter 2

Overview

Webinator is a web walking and indexing package that allows a web site administrator to provide a high quality retrieval interface to collections of HTML and other documents. It is an application of Texis and is written in Texis's Web Script language named Vortex.

It consists primarily of the Texis binary program and two Vortex scripts that are run by the Texis CGI program on your web server and are accessed from a web browser.

One script provides the administrative interface, another provides the site walker and indexer, and the third provides the search function that end users see.

Since these are all scripts, they are easy to modify to provide the look and feel of your site, or to create custom rules for indexing your site.

2.1 Features

Here are some of its features:

- One or more web sites may be indexed into a single database.
- Multiple databases may be maintained.
- It supports cookies.
- There is support for meta data.
- It supports proxy servers.
- Robots.txt and meta robots are respected.
- It provides a totally customizable search interface.
- It provides a totally customizable site walker/indexer.
- A web site may be copied to the local file system.

There are many more features and options to tailor Webinator's behavior to your needs. Almost any option not provided directly by the administrative interface may be achieved by editing the included script(s).

2.2 Obtaining Webinator

Webinator may be obtained from `http://www.thunderstone.com/texis/site/pages/webinator.html`. There you may review the different versions that provide varying size limits and levels of support. Then, you may download the free version or order one of the paid versions.

Follow the instructions on the web site to acquire the package for your operating system. After registering for the free version, you will be given a URL to a compressed tar file for Unix versions or to an executable `.msi` program for the Windows version, and this will contain binaries for your specified operating system.

2.3 Technical Support

Support for Webinator is available via a searchable web message board. It is located at the following URL:

`http://thunderstone.master.com/texis/master/search/msgboard.html`

Anyone may read the discussions. To post a question or comment, you must create an account, which is free, and you must be logged in. Also, once you are signed up, you may "subscribe" to periodic email notifications of new postings to the board. You may select hourly, daily, or weekly notification of new postings.

If you subscribe to periodic notifications, and at some point in the future no longer wish to receive them, you may select "subscribe" again to enter the administrative area where you may delete your subscriptions. Paid users may submit the "Tech Support" form at

`http://www.thunderstone.com/`

under the "Support" link.

Other Webinator resources, such as FAQ, alternate search examples, and such may be found at Webinator's home page `http://www.webinator.com/`.

Chapter 3

Installation

3.1 Linux/Unix Download and Installation

For Linux/Unix platforms, download the `webinator-24.0.1.tar.gz` file, from the URL given to you during the registration procedure, to a temporary directory on your machine. (The number `24.0.1` in the filename may differ, if you are downloading a different version.) Then decompress it, extract it, and run the install script using the following two commands:

```
gunzip < webinator-24.0.1.tar.gz | tar xvf -
cd webinator-24.0.1 && sh ./install.sh
```

Note: The Webinator install must be run as `root`. It will ask for a non-privileged user to own and run the files; this should be the same user your web server uses to run CGI programs (typically a non-login user); consult your web server config files for details (e.g. for Apache this user is typically given by the `User` directive). **Note:** Once installed, Webinator should generally be run as this non-privileged user, not `root`

You will be asked several other questions during the installation. For some of these questions, a default answer may appear in square brackets. E.g.:

```
Install dir [ENTER for /usr/local/morph3]:
```

In this case, if you just hit `Enter` without typing a path, the install will use the answer `/usr/local/morph3` as if you'd typed that. **Note:** Just because a default answer is given, does **not** necessarily mean that is the correct or best answer for your particular environment. It is up to you to choose the default or enter your own value based on knowledge of your machine's setup.

You will be asked the following questions:

- `Install dir`
  This is the directory where Webinator will place its files and subdirectories. It should be a unique (empty) directory. If it does not exist the install script will ask permission to create it for you. The

standard install directory is `/usr/local/morph3`; you should use this if at all possible to avoid potential path issues later. Only enter a different directory if you are specifically unable to install to the standard directory. Whatever directory you choose should be *in*accessible to your web server (i.e. outside its server and document directories): the install will place just the public files of Webinator in your web server tree later.

- `CGI directory`
  This is the directory from which your web server runs CGI programs. The install will create a symbolic link to the `texis` executable here. **Note:** Since Webinator runs as a CGI program your web server **must** be configured to run CGI programs. Consult your web server documentation and config files to find out how and where your server places CGI programs. For Apache servers it is typically done with a `ScriptAlias` directive. Note that this is the *file* path to your CGI directory, not the URL entered in a browser.

- `CGI URL prefix`
  This is the URL prefix to the CGI directory you just entered. In other words, it's the URL that you would enter in a browser to access a CGI program in that directory, but without the program name.

  For example, assume you already have a CGI program `findit` installed on this machine, and you access it via the URL `http://www.example.com/cgi-bin/findit`. You would enter `/cgi-bin` as your URL prefix. If your site uses virtual hosts, or runs on a non-standard port, you can enter a full URL instead (e.g. `http://myothersite.example.com:2001/cgi-bin`).

  If you want to start over with a new CGI directory (previous question), then enter `/newdir` to back up a step.

- `CGI extension`
  This is the filename extension that CGI programs have in the URL. On some web servers, instead of just one directory for CGI programs, any program with a special extension such as `.cgi` at the end signifies a CGI program. If this is true for the CGI URL prefix you've selected, enter it here. For example, if your CGI programs are named `findit.cgi` or `shop.cgi`, then you might enter `.cgi` as the extension. (This may be the case for Apache servers if CGI is set up with an `AddHandler cgi-script` directive instead of `ScriptAlias`.) If your programs do not have an extension in the URL, type `none`.

- `Webinator admin password`
  This is the password for the default Webinator administration account. This password is used to control access to your Webinator walks, so choose a password with care, and ensure that only authorized administrators know it. (Once installed, you can create multiple administration accounts with different passwords if you desire, from the web-based admin interface.) Under some circumstances on some OSes, setting the password from the install may fail. Don't worry: you will be asked to set the password the first time you access the administrative interface.

Once the installation has completed successfully, you can remove the tar and install files, as they are no longer needed:

```
rm -f install webinator-24.0.1.tar.gz webinator.tar
```

Note: If you move your web server directories around or change your CGI configuration after installing Webinator, you will have to re-install it.

Note: To completely uninstall Webinator and all its files, including any walk data, run the uninstaller (located in the install dir) as `root`, e.g.:

```
/usr/local/morph3/uninstall.sh
```

3.2 Windows Download and Installation

The Windows version of Webinator runs on Windows Server 2008/2012 and later 64-bit versions of Windows. Download and run the installation program `setupWebinator-24.0.1-x64.msi` from the URL you were given during the registration procedure. (The number `24.0.1` in the filename may differ, if you are downloading a different version.)

During the install you will be prompted for the following choices:

- **Install directory**
  This is the directory where Webinator will place its files and subdirectories. The directory you choose should be *in*accessible to your web server (i.e. outside its server and document directories): the install will place the public files of Webinator in your web server tree later.
- **Webinator admin password**
  This is the password that the default Webinator administration account will have. This password is used to control access to your Webinator walks, so choose a password with care, and ensure that only authorized administrators know it. (Once installed, you can create multiple administration accounts with different passwords if you desire, from the web-based admin interface.)

3.3 Filesystem Layout

Webinator is installed underneath `/usr/local/morph3` on Unix or `C:\Program Files\Thunderstone Software\Webinator` on Windows by default. It consists of several subdirectories.

This will be the structure on Unix (not all files are listed here):

```
Install Directory
   Readme.txt
   license.key
   texis.ini
   .htaccess
   bin/
      anytotx
      monitor
      texis
   htdocs/
      webinator/
         bar0.gif
         bar1.gif
         common/
            search.css
         factorydefault/
            search.css
         index.html
         xsl/
            default.xsl
   texis/
      default/
         db1/
         db2/
      monitor.log
      scripts/
         errorscript.vs
         webinator/
            dowalk.vs
            search.vs
      testdb/
      vortex.log

HTML Directory
   webinator/
      (same as Install Directory/htdocs/webinator/ tree)

CGI Directory
```

```
texis
```

Note: In versions prior to 6, the configuration file was called `texis.cnf` instead of `texis.ini`. Version 6 will try to load it from the old location if it cannot be found at the new location.

The `webinator` directory contains the search interface scripts, several GIF files used by the search interfaces, and an `index.html` that contains a hyperlink to the administrative interface, as well as the online documentation.

All of the directories that should not be referenced by web browsers contain a `.htaccess` file that denies all access in the event that you chose an install dir under your web server's document root (not recommended). If you did install under your document root and your web server does not respect `.htaccess` style protection you should block web access to those directories by whatever means your web server provides.

This will be the structure on Windows (not all files are listed here):

```
Install Directory
   Readme.txt
   license.key
   texis.ini
   anytotx.exe
   monitor.exe
   texis.exe
   htdocs\
      webinator\
         bar0.gif
         bar1.gif
         common\
            search.css
         factorydefault\
            search.css
         index.htm
         xsl\
            default.xsl
   texis\
      default\
         db1\
         db2\
      monitor.log
      scripts\
         errorscript.vs
         webinator\
            dowalk.vs
            search.vs
      testdb\
      vortex.log

CGI Directory
   texis.exe
```

The `bin` or install directory contains the `texis` program and other related utility programs.

The `texis` directory contains the databases and Texis log files.

3.4 File Permissions and OS Specific Notes

- **Windows**
  IIS will typically run `texis.exe` as the anonymous user `IUSR_machine`. If you want searches to automatically recompile scripts for you, then this user will need write permission on the directories containing the scripts: `texis/scripts/webinator`.

Another option is to test and compile the scripts in a staging area, and when you are satisfied with the results, simply move the compiled `.vsc` file into place.

Texis requires that its monitor process is running. It will attempt to start it if it's not already running. When Texis is running under the web server, there might not be permission available for it to run properly. As administrator, you can register the Texis monitor as a service to run in background and when the system starts up. The install will do this if run as an administrator. You can do this manually from a command prompt when logged in as administrator:

```
monitor -R
```

This will start the monitor service immediately, so there's no need to reboot to activate it.

If you ever wish to unregister the Texis monitor as a service, do this from a command prompt when logged in as administrator:

```
monitor -U
```

- **Unix**
  It is important that texis and its related utility programs always run as the same userid, and that that userid is the owner of the databases. Web servers generally run CGI programs as some user with little or no permission. The installation attempts to get around this problem by making the programs `setuid` to the correct user. If it is not able, you will receive a warning. It is up to you to ensure that `texis` is always run as the same userid.

  The standard Unix commands for making a program `setuid` to some user, `myself` for example, are:

  ```
  chown myself texis
  chmod u+s texis
  ```

  The above commands may only be run by the `root` user on some systems.

3.5 Customizing Webinator's Appearance

You may make common changes to Webinator's search appearance by using **Search Settings** from the administrative interface main menu. You may select color, font, size, results style and order, as well as setting boilerplate HTML to wrap around the search form and results.

But you are not limited to these features. You may change any and all aspects of the search program's appearance and behavior by modifying the supplied `search` script or writing a completely new one.

See

`http://docs.thunderstone.com/texis/site/pages/webinator_extras.html`

for some examples of custom scripts.

For details on programming with Texis Web Script (Vortex), see the Vortex manual at the Thunderstone web site, `http://docs.thunderstone.com/site/vortexman/`.

See also **Customizing the Search** (section 6.7, p. 206) for some insight into the inner workings of the default search script.

Chapter 4

Operation

4.1 Running the Administrative Interface

Webinator's administrative interface is a web application that you access using your web browser. Access it using the URL that was given to you during installation. It will be something like:

- On Unix:
  `http://YOURSERVER/cgi-bin/texis/webinator/dowalk`
- On Windows using CGI:
  `http://YOURSERVER/scripts/texis.exe/webinator/dowalk`
- On Windows using ISAPI:
  `http://YOURSERVER/texis/webinator/dowalk`

Where `YOURSERVER` is the hostname, and possibly the port number, used to access the web server where Webinator is installed.

The `cgi-bin` and `scripts` portions refer to the CGI directory you specified during installation. The examples given above are the most common. Your path could be different.

`texis` and `texis.exe` are the names of the Texis Web Script interpreter and is a program that resides in your CGI directory. It is not a directory.

The portion after `texis`, `/webinator/dowalk`, is a "virtual" path indicating the location of the administrative script relative to your installation's `ScriptRoot` directory. `ScriptRoot` is the `texis/scripts` subdir of your install, so `/webinator/dowalk` in the URL is referring to the file `texis/scripts/webinator/dowalk` under your install dir. This is the administrative script that controls Webinator.

When you run the administrative interface you will be asked for the login and password. By default there is one login name. It is `webinator` in all lowercase. If no other accounts have been added, you will not have to enter the name. It will be filled in for you. Your login will be remembered in a cookie until you logout. This way, you don't need to enter the password every time you enter.

Note: If you share your computer with others, or it is available to people who should not be administering Webinator, then you should logout when you are finished. This will help prevent unauthorized configuration.

The Webinator administrative interface uses JavaScript to enhance its functionality and make it easy to use, but the interface will also work well without JavaScript. No functionality of Webinator will be lost if JavaScript is turned off in your browser (e.g. to prevent pop-ups on other sites). In this document, the user interface description assumes that JavaScript is enabled.

4.2 First Time Run: Quick Start

Step 1: Create an Account

During installation you were asked for a password for the default administration account (`webinator`), which you should now enter at the prompt. If for some reason this step did not happen, the first time you run the administrative interface you will be asked to create and enter a password. You should choose a password that is easy for you to remember but hard for someone else to guess, as this is an account that will control administrative access to Webinator (additional accounts may be created later as needed). You will need to enter the same password twice (two input boxes will be provided) to help check for typing mistakes. Passwords are case sensitive. Once the password is created and `Change` is pressed, you will automatically be logged in and taken to the `Profiles` page to create a profile.

Step 2: Create a Profile

A *profile* is a collection of data (URLs/documents) to be searched, plus the settings that control that search; a profile must be created and walked before searches can occur.

On the `Profiles` page, a default profile name and data directory may already be filled in for you to create. If **Profile Dataspace Roots** have been configured in **System Wide Settings**, you'll have a set list of data directories to choose from.

Enter a name for the new profile, and choose a profile type. A `Standard` profile is just that – a standard profile for walking – and is usually what you'll want to create, especially for the first profile. A `Meta Search` profile does not walk data itself, but merely searches and aggregates results from one or more other profiles; see p. 132 for details. After setting a profile name and type, hit the `Create Profile` button to create the profile.

A new profile will be created but a site walk/index will not be started yet. You are then presented with the main walk settings page. Use the `Base URL` setting to specify the starting point of your walk. This is often the homepage of a site, or the sitemap page.

Step 3: Walk the Profile

Once you're satisfied with the URL and extension settings, you may hit the `GO` or `Update and GO` button to begin a walk of your site. A walk will be started in the background and you will be taken to the `Walk Status` page. This page will show you the status of the walk in progress and indicate when the

walk is complete. This page will automatically refresh every 10 seconds with the latest progress information until the walk is complete. When the walk is complete you will see a summary of errors.

Last Step: Search

Once the walk is complete, you may click `Live Search` on the menu at the top of the page. This will take you to the search that users will use. It is also the URL you can place on your web page(s) to send users to the search.

You now have a site index that you can use. There are many options to control the site walk as well as the search interface appearance. They are described in detail elsewhere in this manual. Use the `All Walk Settings` button on the administration script's menu to see all of the options. Click the question mark (`?`) next to an item to get help for that item.

Since the walker, administrative interface, and search are all scripts with source code provided, you are not limited to the settings available in the administrative interface. Any or all of the scripts may be modified to take on new behaviors.

4.3 Administrative Interface Overview

Webinator's administrative menu has the structure given below. Each item is described on the pages that follow.

```
Settings
  Basic Walk Settings
  All Walk Settings
  Search Settings
Tools
  List/Edit URLs
  Browse URLs by Folder
  List Duplicates
  Test Fetch
  Test Search
  Query Log
  Replication Tools
  SOAP Tools
  Integration Tools
  Best Bet Groups
Status
Search
Profiles
System
  Information
    System Information
    Document Usage Overview
    Test Network and Servers
    Task Monitor
  Modules
    Thesaurus
    Client Certificates
    OneBox Providers
  System Setup
    System Wide Settings
    Apply a License
  Backup/Restore Settings
    Backup Webinator Settings
    Restore Webinator Settings
  System Replication
    System Replication Queue
    System Replication Target Status
  Security
    Accounts & Groups
    Access Control Lists
  Advanced Tools
```

```
    Repair Tools
      Check Version Upgrade Actions
      Re-output XSL files
      Re-schedule walks
Docs
```

Basic Walk Settings

This is the central area for configuring a walk. The most commonly used walk related options and their settings are presented and they may be changed here. The Basic Walk Settings are a subset of the All Walk Settings. Next to each option is a question mark (`?`) which, if clicked, takes you to help for that option. The options are documented individually later in this manual in section 4.4.

At the bottom of the page is a set of three buttons. Pressing any of the buttons affects all options on the entire page.

- `Update`
  This button causes all changes on the form to be saved. No walk is started.

  If the **Rewalk Schedule** has been changed, the new schedule will go into effect immediately.

  If **Categories** have been changed, the walk database will be updated to reflect the new categories. The search interface will reflect the new categories.

  If **Single Page**, **Page File**, or **Page URL** has been changed, the listed individual pages will be fetched into the live search database and made available for searching.

  If the **Word Definition** or **Text Search Mode** is changed, the search index on the live database will be dropped and recreated. Searches might not work while the index is being rebuilt.

- `GO` or `Update and GO`
  The `GO` button will change to `Update and GO` after you make a change to any setting on the form. The ultimate behavior for either is the same.

  The current settings from the form will be saved as is done when you click `Update`. Then a new walk will be started. The new walk will be performed to either a temporary database or the live database, depending on the setting of Rewalk Type (Section 4.4.14). Then you will be shown the walk status page where you may monitor the progress of the walk.

  Changes to **Categories** or **Word Definition** will not be reflected until the walk finishes.

- `STOP`
  When a walk is in progress the `GO` button is replaced by the `STOP` button. This button terminates the running walk and abandon the work that it has done so far.

- `Reset`
  This button reverts all settings on the page to what they were when the page was first loaded.

All Walk Settings

This is the central area for configuring a walk. This is similar to `Basic Walk Settings` except that all walk related options and their settings are enumerated and may be changed here. Please see section 4.5 for details on the individual settings.

Search Settings

This page contains all of the settings related to the search interface that end users see when performing searches.

All search options and their settings are enumerated and may be changed here. Next to each option is a question mark (`?`) which, if clicked, opens help for that option. The options are documented individually later in this manual in section 4.6.

At the bottom of the page is a set of buttons. Pressing any of the buttons affects all options on the entire page.

- `Update Test`
  This button causes all changes on the form to be saved in the set of test settings, which can be tested via the `Test Search` link on the left side of the interface. It does **not** modify the `Live Search` settings. This allows you to "try out" settings before applying the changes to your live search users' interface.
- `Update Live and Test`
  This button updates both the `Live Search` and `Test Search` settings. Use this either after testing out the settings via `Update Test`, or for small changes that you don't feel the need to test out and immediately want to make live.
- `Copy Live to Test`
  If you try out changes via `Test Search` and you decide you don't want them, you can use `Copy Live to Test` to discard the test changes you've made and revert back to the current `Live Search` settings.
- `Reset`
  This button reverts all settings on the page to what they were when the page was first loaded.

List/Edit URLs

On this page, you may list or delete all or selected URLs from the database. You should always list before you delete, so you know that you are deleting the correct ones. While listing URLs, you may display all known information about a given page. You may also create categories for selected sets of URLs from this interface.

If a walk is in progress, delete is disabled and you are given the choice of listing URLs from the live search database or the new database being built by the walk.

Select `List` or `Delete` from the drop down list. The default is always `List` for safety.

In the pattern box, enter the URL or pattern for URLs for which you want information. This may be an exact URL or a wildcard pattern, which lists all URLs matching the wildcard pattern. For a wildcard pattern, use asterisk (`*`) to match anything and question mark (`?`) to match any single character. You may enter up to 10 different URLs or patterns in the box to find them all at once. Put a space between patterns when entering multiples. Leaving the pattern box blank implies `*`, and this will cause every URL in the database to be listed. Deletion will be denied if the pattern is blank or `*`.

Select the order in which you wish to see the list:

Depth	URLs encountered first in the walk will be listed first
URL	URLs are ordered alphabetically
Newest first	URLs are ordered by modification date with newest ones first
Oldest first	URLs are ordered by modification date with oldest ones first
Largest first	URLs are ordered by download size with largest ones first
Smallest first	URLs are ordered by download size with smallest ones first

Then `Submit`.

All matching URLs will be listed. Clicking on a listed URL opens a page of details about that URL. On that detail page, everything the database knows about that URL is presented. You can also see what pages refer to the selected page by clicking `Parents` and what pages the selected page refers to by clicking `Children`. The `test` link next to the URL can be used to do a live test fetch of the page to find out how Webinator processes it. See `Test Fetch` 4.3.

If your pattern matches less than the entire database, you will be given a form from which you can create a category using the same pattern(s). Simply enter the name of the category to create and click `Submit`. The name is the name that users will see on the search form. This new category will also appear on the main settings page along with the other categories. It will also be immediately available to search users.

If the profile is a meta search, then the profile has no URLs of its own to list. The List/Edit URLs page will instead display links to the list/edit URL pages for each of its target profiles.

Live Search and New Walking Database

These options are presented on the `List/Edit URLs` page (see 4.3) if a walk is active. They allow you to choose which database to query. The “Live” database is the one from a previous successful walk that is what search users see. The “New” database is the database currently being built by the new walk. It is not visible to search users.

Browse URLs by Folder

Browse URLs by Folder allows you to view the contents of the profile by folder. You can see the total number of items that exist within a given folder, regardless of which pages link to which other pages.

Clicking on a folder descends into that folder, listing its contents. Clicking on a file takes you to its `List/Edit URLs` page.

List Duplicates

This section allows you to list all the duplicates of a given page. The URL entered may be the URL that was kept in the walk, or any of the pages that were excluded as a duplicate of pages already in the walk.

If `Keep Refs` was used in the walk, then all the pages that linked to the duplicate pages will also be listed.

Test Fetch

This allows for testing Webinator fetching of URLs. Be sure to properly encode any entered URL, like space as `%20`.

Several processing options are provided to control how much processing to do. Expand the `Options` link to show and edit these options. Note that most will only be set at the request of Thunderstone tech support. Some options may produce copious messages.

- `Full Processing`
  Perform full processing on the fetched file as if it is being prepared for the search database (execute any relevant Primer URLs before-hand, apply rejection rules to its links, etc). Otherwise only perform the basic download of the page.
- `Keep Download`
  Keep the raw encoded download and decoded data for display. Using this can make the test results page particularly large for large source documents like PDFs etc.
- `No redirects`
  Do not follow redirects. This can be useful to get the full size, content etc. of a page in a redirect chain. When checked, **Max Redirects** (p. 92) is set to 0 for the fetch.
- `Trace Settings`
  Defaults to, and overrides, the **Walk Trace Settings** option (p. 87): a set of zero or more comma-separated “name=value” pairs to generate additional debug/trace messages; set at the request of Thunderstone tech support.

A short summary will be shown followed by various statistics and other information about the page. Most of the information is collapsed (hidden) to reduce page clutter. Click the + next to an item to expand that item for viewing. Click the − to collapse an item. Use the `Collapse all` and `Expand all` links to Collapse all items or expand all items respectively. Use `Show empty fields` to show all fields even if there was no data for them. That helps one determine that a value is actually missing as opposed to overlooked for display.

Large text fields will be shown in scrollable areas by default to avoid taking over the page. Click the + next to a scrolling area to let it fully expand onto the page. Click the − to confine and expanded field.

Test Search

This hyperlink opens the search interface. It forces the interface to use the search settings listed on the `Search Settings` page, whether they have been applied to the Live Settings or not. This allows you to

test search settings without affecting end users until you are satisfied with the new settings.

This mode also places two extra hyperlinks at the top of the search pages. `Back to Administration` allows you to return to the Webinator administration interface. `Make this appearance live` does that too, but it additionally makes the search settings you are testing "live", so that end users also see the search setting effects.

Query Log

The query log pages provide detailed and summary information about queries. Query logging must be turned on to generate information on the query log pages. If query logging has never been turned on for the current profile, there will be nothing to see. The query log is erased each time the database is rewalked.

The pages are as follows:

- Query Report
- Top Query Words
- Top Queries
- No Hits
- Best Bet Clicks

The query log lists the time that each search occurred, the IP address of the web user performing the search, the number of hits for the search, and the user's query. For result clicks, it displays the query instead of the number of hits and the actual URL instead of the query.

Selecting the Date/Time for a listed query will display a page with complete information about the search. This page includes everything from the summary list, and any non-default parameter settings from the search. A hyperlink is provided so that you may perform the same query as the user.

Administrators can use the "from" and "to" widgets to restrict the date ranges used to generate the reports.

Note: If the search user is going through a proxy that provides the `X-Forwarded-For` HTTP header, that forwarded IP will be logged as the search user's IP and the proxy's IP address will be logged as a `ForwardedBy` value.

Replication Tools

Replication Tools allows you to work with replicating data from this profile. *Replication is supported in the full Texis product, but not Webinator-only.*

- **View Replication Status**

  Replication Status shows you an overview of the contents of the replication queue. The data is presented grouped by host/profile, and the next items queued are detailed below.

- **Send Profile Settings**

  Send Profile Settings is used when you want to send this profile's configuration to another Webinator, possibly with a different name. If the specified profile doesn't exist on the target machine, it will be created and given this profile's settings. If it already exists, it will have its settings set to this profile's values.

  Enter a remote machine and profile name & hit `Send Settings` to queue up the Send Settings action in the replication queue, which you can view with the previous link.

- **Send Profile Data**

  Send Profile Data allows you send all the current profile's data to an established replication target. This can be useful when adding a new target to existing sender profile, and you need to load the target with existing content but don't want to perform a full walk on the sender.

  Select one of the machine/profile pairs listed, and hit `Send Data` to queue all this profile's content for replication. Please see the Replication Status page describe above to monitor the progress.

SOAP Tools

Soap Tools provides various WSDLs and references for working with Webinator via the SOAP API. Please see the SOAP API section for more details (5.20)

Integration Tools

This page provides tools for integrating the Webinator search interface into your own site. This is related to where you want your search interface to appear, which may be separate from the content that is being indexed.

The Javascript search dropin defines a block of static HTML that can be used to place a search interface for the current profile within any other HTML page.

Copy and paste the HTML code within the gray block into any HTML page, and javascript will place a full search interface within that page. Submitting searches or following links to further search results will stay within that page. Any look and feel customizations applied in the profile will still apply.

Search forms in your own page can use the `ThunderstoneForm` class, which cause their content to be updated with the search. For example, if a user types in a search for `tset` and clicks on the Spelling Suggestion `test`, this will allow the `query` box to be updated with the now-current query, `test`.

Note: The Javascript dropin does not support results authorization.

Best Bet Groups

The Best Bets are grouped together. This allows different groups to be shown in different places, and easily rotated in or out. For example, you might have one group of links that you have determined to be the most probable results for a user's query, and another group that includes links you want to promote.

The Group Name is how the group will be identified elsewhere in the administrative interface. This should be chosen to readily remind you of the purpose behind the group.

The Result Type indicates which fields will be shown on the results page. The title and description are entered by the administrator, rather than always being taken from the page.

Status

This page shows the status of the latest walk for the current profile. If a walk is in progress, it is the one reported.

During an active walk, it indicates a summary of how many pages are to be walked in the next hour, how many were walked in the last hour, and the total number of pages. There is a list of the most-recent URLs fetched, with number of errors and duplicates found, followed by a list of the next URLs to be walked. Below that is summary information about the walk itself, including walk start time, starting URLs, and some profile settings. The Walk Status page updates automatically every 10 seconds until the walk is complete or another page is selected. (After 10 minutes of user inactivity it will refresh once a minute to save traffic.)

When no walk is in progress, the report also includes a list of errors and duplicates encountered. If the last walk was abandoned, the report includes information about how far it went, as well as the report from the last complete walk.

Walk Status tabs

If more than one database is available for viewing, tabs will appear at the top of the Walk Status, allowing you to swap between database. This can be caused by a ”New” crawl running (allowing you to switch between `New Walk Database` and `Live Search Database`), or if a ”New” crawl failed and automatically reverted to the previous database (allowing you to switch between `Live Search Database` and `Failed Walk Database`).

If more than one walk has been performed, then an `Archived Logs` tab will be available. This lets you view log files from previous crawls for errors or other unexpected behavior. No database or searchable content from these old walks is retained, only the log files.

Old archived logs are automatically cleaned out if they become too numerous or consume too much space, with the oldest logs removed first.

While a walk is occurring, multiple buttons are available:

- **Now button**

  During the walk the **Refresh display:** `Now` button may be selected to force a Walk Status display refresh before the 10 second automatic refresh. Note that this only affects the display, not the walk itself.

- **Pause/Auto button**

  The **Refresh display:** `Pause` button pauses the Walk Status display (prevent the browser from refreshing the display every 10 seconds): this changes the button to `Auto` which will have the opposite effect (resume the auto-refresh). This is useful when examining the status page in detail, and

avoiding being interrupted by the browser auto-refresh. Note that both buttons only affect the display, not the walk itself.

- **STOP walk button**

  The **Current run:** `STOP walk` button on the Walk Status page stops the current walk. If the walk type is `New`, the walk will be abandoned (current live search is left intact and not updated). If the walk type is `Refresh`, the new pages are always live (since refresh uses one database), but the search indexes are not updated.

- **Pause walk and Make live button**

  The **Current run:** `Pause walk and Make live` button pauses the current walk, updates its search indexes for speed, and makes the walk live (i.e. deletes the current live database and replaces it with the current walk). This can be useful if you ran out of disk space while indexing and subsequently freed up some space, or if a long running walk was stopped and you want to use the incomplete walk. If the walk was abandoned due to an error, make sure you resolve the problem before trying to make the new database live.

Search

This hyperlink opens the Webinator search interface as end users see it.

Profiles

This page presents a list of existing profiles. A profile contains the walk and search settings for a collection of pages. The profiles are listed in the order of creation by default; clicking on `Name` will re-order by profile name. You can click on a profile's name to see and/or change its settings and status or to start a walk.

You can click on `Delete` next to a profile to delete that profile. You will be asked whether you really want to delete the profile or not.

When a profile is deleted, all of its settings are lost and any walk database it has created is deleted. There is no way to get back any of these items after the profile is deleted. **Note:** Under Windows it is possible that the walk database will not be completely deleted if there are currently searches being performed on the database. You should not delete a database that is being actively searched. If you do this, you will need to delete the remnants of the database by hand.

You may also create a new profile by entering a new name and data directory.

If `Profile Dataspace Roots` has been configured, you'll have a set of dataspace prefixes to choose from. Your profile dataspace will be created as a subdirectory of the chosen root.

Otherwise, you'll have a freeform text entry for your data directory. You may not use a data directory that is in use by another profile. You generally specify a nonexistent directory. The directory will be created if it does not already exist.

You can copy settings from an existing profile to your new profile by selecting its name from the drop down list. This allows you to set up another site similar to an existing one. It allows you to experiment with the

walk settings for an existing site, without potentially harming the good walk that is being searched by your users.

System

The System section contains various links for maintaining and editing operating-system and overall settings.

System Information

This page provides general system information, such as Version details and Thunderstone license information.

Document Usage Overview

Document Usage Overview shows how much your combined profiles are using of your purchased license. The total number of results of all profiles are added together, and displayed in a doughnut chart to show which profiles are larger than others.

Test Network and Servers

This area provides the ability to test the network connectivity of Webinator and find what web and file server documents look like to it. It is divided into two sections. The first section is for testing Webinator fetching and processing of urls. The second section is for testing Webinator's general network connectivity.

- **Test URL fetch**

  This is the same functionality as found in `Test Fetch` in a profile's `Tools`. Please see its documentation (p. 36) for more detail.

- **Test Network**

  There are several network tests available. As many as desired may be done together. Each will be executed in sequence one after the other and the results presented together on one page.

  - Find IP
    Look up an IP address for a given host. Options (correspond to walk DNS Mode settings):
    - Internal - Perform the lookup using internal parallelizing routines.
    - System - Perform the lookup using standard system routines.
  - Ping
    Send ping packets to the given hostname or IP address to determine reachability and speed. Check `Gateway` to ping the configured gateway address. A handful of packets will be sent and statistics about each and a summary of response times and loss will be displayed. *Note that not all machines respond to ping and some firewalls block ping. Page fetching may still work even if ping doesn't.*

- Traceroute
  Trace the network route to the given hostname or IP address to determine reachability and spot possible problem areas. It will display one line for each hop along the network route to the target machine. Asterisks (*) indicate a problem finding the next hop. *Note that some firewalls and routers block traceroute. Page fetching may still work even if traceroute doesn't.*
- Email
  Send a small test email to the given email address. This will test Webinator's email configuration as well as the recipient's ability to receive emails from Webinator. If the recipient doesn't get the test email look in `System` → `Manage Logs` → `maillog`' to see if the message was handed off successfully. If it was handed off check the recipient's spam folder.

Task Monitor

Provides an interface to manage the Task Monitor, which is a background/daemon process that automatically handles various Webinator tasks, e.g. updating indexes during settings replication. The queue of pending tasks can be viewed and/or deleted (only if needed by tech support).

Thesaurus

This area allows you to upload one or more custom thesauruses (synonym lists) for use by search profiles. An uploaded thesaurus is compiled and kept on Webinator. You can download the thesaurus later by clicking on it in the listing.

Each thesaurus may be used by zero or more profiles and should not be deleted if it is in use by a profile. Search options that affect the use of these thesauruses are `Synonyms`(4.6.62), `Main Thesaurus`(4.6.63), and `Secondary Thesaurus`(4.6.64).

See section 5.3 for further details.

Client Certificates

This area allows you to upload one or more client certificates to use while authenticating with HTTPS servers. These are normally not needed unless the remote server requires a client certificate for authorization.

Adding a new certificate requires providing the certificate and key each in PEM format (with the `BEGIN PRIVATE KEY/END PRIVATE KEY` and `BEGIN CERTIFICATE/END CERTIFICATE` blocks, respectively).

Client certificates uploaded here can be chosen for use with the `Client Certificates` profile setting (4.5.67). The same client certificate can be used by multiple profiles.

OneBox Providers

Experimental This area lets you configure OneBox Providers.

This is currently an experimental feature which may or may not be available in this product.

System Wide Settings

This area is for settings that affect Webinator as a whole and/or may be shared by multiple walk profiles. Please see section 4.7 for documentation on the individual settings.

Apply a License

This allows a license update (obtained from Thunderstone) to be applied to Webinator, to upgrade features or increase limits. A form is provided to accept the license and verify credentials. Only the `webinator` user can apply a license update, and that account's password must be given again on the form for security. The license is provided either by selecting the file it was saved to via the "`License from file`" box, or cut-and-pasting it (e.g. from an email) into the "`Or copy license text`" box.

Hit "`Apply License`" to apply the license. If the license was installed successfully, the message "`License applied successfully`" will be shown. If it could not be installed, an error message will be shown.

Note: A full install or upgrade (not just scripts) to Webinator version 6 or later is required for this feature. Also, it must have been enabled in `texis.ini` via the `[License Update] User` setting (enabled by installer).

If the license could not be installed, the alternate method is to copy it to a file named `license.upd` in the install directory (typically `/usr/local/morph3` under Unix or `C:\Program Files\Thunderstone Software\Webinator` under Windows), then from a command prompt, cd to the install dir and run `texis -update` (Windows) or `bin/texis -update` (Unix) to apply the license.

Some typical errors that might occur when installing a license from the web form include the following:

- `License update unauthorized`
  The password may be incorrect, or the currently-logged-in user may not be permitted to update licenses (usually must be `webinator`).
- `Invalid license`
  The license supplied was invalid or unacceptable. Make sure it is a valid Webinator license, not a third-party license (e.g. for the Language Analysis Module). Make sure it was cut-and-pasted cleanly from any email message it was embedded in.
- `Service Not Enabled`
  License updates via the web admin interface were not enabled at install/upgrade time. Use the command-line interface (above).
- `Secure Connection Required`
  A secure (SSL) connection from Webinator to the Texis Monitor process (which manages licenses) could not be established. Use the command-line interface (above).

- `Internal error`
  An internal Texis Monitor error occurred. Use the command-line interface (above).

Backup Webinator Settings

This allows you to save all of the current profile and most of the system settings from Webinator to an XML file on your local workstation. This file can be used to aid in cloning Webinators for a cluster and as a backup in the event the machine needs to be restored from scratch.

"System-Wide Settings" includes things not specific to a profile - admin logins, system-wide settings, etc. You can choose to download the settings for all profiles, or for some combination of profiles.

"Internal Settings" includes things that aren't used when restoring, such as currently running Process IDs. This should only be included at the request of Thunderstone Support.

Click `Download` to save a copy of the current settings to your workstation.

Restore Webinator Settings

Use this option to restore settings that you've previously captured using `Save Webinator settings`. Missing profiles will be created, and existing profiles will have their settings set to the values contained in the backup.

System Replication Queue

Provides an interface to view and manage the queue for "non-profile" content when using System Replication Settings. Non-profile content is things that do not apply just one profile, such as `System Wide Settings`.

System Replication Target Status

Allows you to check the status of all the profiles on this sender machine against all System Replication targets at once, indicating which profiles are present and which aren't. It also allows you to create the missing profiles on the targets.

Accounts & Groups

This section provides information to maintain multiple login accounts for access to Webinator administration. All users are listed on this page. You may add users, delete users, and change individual user passwords. The default user, called `webinator`, may not be deleted.

The Accounts page also allows you to create multiple administrative users. There is no distinction among them after they are created. All users have full administrative permissions, and they may create and delete any user or change any user's password. This is a basic security mechanism meant to keep unauthorized

persons from using the web based administrative interface. The purpose of supporting multiple administrative users is that you can create distinct passwords, which you can revoke in the future without needing to change a single global password that all administrators know.

User names and passwords are stored in the `SYSUSERS` table of the default database. This is only a holding place for them. No Texis permissions are granted or revoked for these users. A benefit of storing the users in `SYSUSERS` is that any users that you might create in the default database by other means than the Webinator interface will also automatically become Webinator administrators.

Usernames may only contain letters, numbers, and underscores, they must begin with a letter, and they must be 20 characters or less. Names and passwords are case sensitive.

The passwords are one-way (forward) encrypted. This means that a forgotten password may not be discovered. The only way to deal with a forgotten password is to change the password. In the event that all passwords are forgotten you can delete the `webinator` user from SYSUSERS using `texis -s` from a command prompt, and then enter an appropriate SQL delete statement. The administrative script will then create the `webinator` user anew and ask you for a new password.

User groups may be created on this page, by clicking the `Add a Group` link. Existing groups may be edited or deleted with the appropriate links. User groups are used to associate administrative users into similar-privilege groups for easier access control maintenance. See the User Groups section for more details (p. 134).

User groups are supported in the full Texis product, but not Webinator-only.

Access Control Lists

The Access Control page allows configuration of administrative users' access to administrative actions (creating profiles, starting walks etc.). In conjunction with user groups, access control can be used to restrict certain users to only certain actions, instead of allowing all users access to all administrative functions. See the Access Control section for more details (p. 134).

Access Control is supported in the full Texis product, but not Webinator-only.

Repair Tools

Provides internal verification (and any necessary fixes) of Webinator systems, including actions automatically taken at upgrades.

These are tools are meant to address situations that the Webinator should not normally find itself in Only use at the request of Tech Support.

Check Version Upgrade Actions

There are internal actions automatically performed when the Webinator upgrades between major versions, such as creating new database tables. If something unexpected went wrong in the upgrade process, this section lets you check for and re-apply the upgrade actions. This is usually not needed, and should only be

done under the advice of Thunderstone Support.

Re-output XSL files

In the past, when a profile was restored from backup or made as a copy, it was possible for a profile's XSL files on disk to become out of sync with the profile's settings. This has been fixed, but customers that had restored profiles using old software may have profiles in this state.

This tool checks which profiles have XSL settings and files, and will re-write the profile's XSL data to disk if necessary. Not normally needed.

Re-schedule walks

In the past, when a profile was restored from backup, it was not actually scheduled with the profile's rewalk schedule setting. This has been fixed, but customers that had restored profiles using old software may still have profiles in this state.

The `Re-schedule walks` section confirms that all profiles' schedules match their rewalk schedule settings, and allows the re-application all profile's scheduled settings to the walk scheduler. Not normally needed.

Docs

This provides a hyperlink to the online version of this documentation. It also contains a link to download a PDF version of this documentation. `http://www.thunderstone.com/site/webinator5man/`

4.4 Basic Walk Settings

This page contains the settings that are used most commonly. They are available in `Basic Walk Settings`.

The settings on the `Basic Walk Settings` page are a subset of the settings on the `All Settings` page. Use the page that is most convenient for your current task.

4.4.1 Database

Syntax: the full path to the database directory on the server's disk

This indicates what database is being used by the currently selected profile. The database is only settable when creating a profile. A new profile must be created to use a new database.

4.4.2 Walk Summary

This is informational only. It contains summary information about the most recent walk, and any recent recategorization (see **Categories**, p. 55). The information includes the date and time of the walk, whether the walk was successful, how many pages were indexed, and the number of duplicate pages.

4.4.3 Notes

This is a scratch pad area for the administrator of the profile. It in no way affects the walk or search.

4.4.4 Base URL(s)

Syntax: one or more URLs, one per line

This is the address where the web walker will start walking your site. If the whole site is to be searched, simply enter your web address, for example "`http://www.example.com`". If the search is to be limited, specify the address to start the search or create a page listing the URLs to search. The search will only return information from your web site - no off-site searching will be done. Directory URLs should include a final forward slash "`/`". Example - "`http://www.example.com/mysite/`". If you have a virtual domain that just redirects to another URL, enter the destination URL as your Base URL instead of your virtual domain name.

You may specify multiple base URLs to index multiple sites; Webinator's idea of a "site" is a single host as identified by the hostname portion of a URL. Therefore `http://www.example.com`, `http://www2.example.com`, and `http://example.com` would all be considered different sites.

In version 4.02.1046373961 Feb 27 2003 and later, the special "protocol" `http-post` or `https-post` may be used for a Base URL. This uses the POST method instead of the GET method to fetch the URL, using the query string as POST data (it must be URL-encoded). This can be used to start walking at a login page form that requires POST instead of GET. Note that the URL stored in the `html` table will have the `-post` and query string removed for security. During a `Refresh` walk, when a URL is about to be refreshed, the probable Base URL that led to it (i.e. the one with the longest prefix) will also be fetched. This helps ensure that login cookies are properly restored to allow Webinator access during the refresh. Example:

"`http-post://www.somehost.com/login.asp?user=bigbird&pass=open-sesame`"

See also `URL file` 4.5.6, `URL URL` 4.5.7, `Single page` 4.5.8, `Page file` 4.5.9, and `Page URL` 4.5.10 for more ways to specify URLs.

4.4.5 Robots

Syntax: select Yes or No buttons

robots.txt

With this set to Yes, Webinator will initially get `/robots.txt` from any site being indexed and respect its

directives for what prefixes to ignore. Turning this setting off is not generally recommended. Supported directives in `robots.txt` include `User-agent`, `Disallow`, `Allow`, `Sitemap`, and `Crawl-delay`.

Note that any `Crawl-delay` value will be modified to fit in the **Robots Crawl-delay** range (p. 48, and overrides **Walk Delay** (p. 49.

Any `Sitemap` links in `robots.txt` will be walked as well, subject to normal exclusion settings. Sitemaps not in `robots.txt` may be added via **Base URL(s)** (p. 47) or **URL URL** (p.56).

Meta

When set to `Y`, Webinator will process and respect the meta tag `robots` within each retrieved HTML page. This tag contains per-page robot (walker) control information; see p. 146 for details on its syntax.

Placeholder

Whether to still put an (empty) entry – a placeholder – in the `html` search table for URLs that are excluded via `<meta name="robots">` tags. Leaving a placeholder improves refresh walks, as the URL can then have its own individual refresh time like any other stored URL. Without a placeholder, the URL would be fetched every time a link to it is found, because no knowledge that it has been recently fetched would be stored.

The downside to placeholders is that if the URL is also being searched in queries – i.e. `Url` is part of **Index Fields** – then the excluded URL might be found in results. Placeholders have empty text fields (e.g. no body, meta, etc.) to avoid matches on text, but the URL field must remain.

See also `Robots.txt` 5.4.

4.4.6 Robots Crawl-delay

Syntax: decimal number

This gives the minimum and maximum `robots.txt Crawl-delay` values to allow; values found outside this range will be changed to the appropriate minimum or maximum. -1 means no limit. The defaults are 0 and 10. These values can be used to set reasonable bounds to sites' `Crawl-delay` values.

Note that a `Crawl-delay` seen (modified to these limits) is only used if **Robots robots.txt** is `Y`, and overrides **Walk Delay** (p. 49). Thus, to use the greater of `robots.txt Crawl-delay` or **Walk Delay** (e.g. if the walker is bandwidth-limited, and sites' `robots.txt` delays are to be followed), keep **Min** equal to **Walk Delay**. To always use **Walk Delay** but still respect other `robots.txt` directives (i.e. just ignore `Crawl-delay`), keep **Min** and **Max** equal to **Walk Delay**.

4.4.7 Allow Extensions

Syntax: one or more file extensions separated by space

A list of the URL path extensions that the walker will accept. The default list is empty, and indicates that all extensions are allowed. Include the "`.`" in each listed extension. Case is always ignored. URLs with no extension are always allowed.

E.g. to accept MS-Word documents, add `.doc` to the list. Note that if the list is non-empty, any extensions not listed will not be walked.

A few other potentially useful extensions:

```
.asp
.cfm
.jsp
.shtml
.jhtml
.phtml
```

4.4.8 Exclude Extensions

Syntax: zero or more file extensions separated by space

A list of URL path extensions that the walker will reject. The default is empty, i.e. no extensions will be rejected. Include the "`.`" in each extension. Case is always ignored.

4.4.9 Exclusions

Syntax: zero or more strings, each on a separate line

Excludes URLs containing any of the specified literal strings anywhere in the URL (hostname, path, or query).

See also `Exclusion REX` 4.5.20 and `Exclusion prefix` 4.5.21 for more ways to exclude URLs.

4.4.10 Walk Delay

Syntax: a decimal number from 0 to 10

Causes Webinator to wait the specified number of seconds between page fetches. Normally set this to 0, and Webinator will fetch and process pages as quickly as it can. Increase the **Walk Delay** if the web server cannot handle being hit rapidly. Increasing this value forces the walk to take at least the following number of seconds to complete: the **Walk Delay** number times the number of pages on the site.

Decimal numbers may be specified - `0.1` will cause it to walk no more than 10 pages per second, etc.

Note: Using a delay larger than 0 forces **Threads** (4.4.11) to 1 to avoid possible fetch timeouts. Thus a non-zero delay defeats the advantage of multiple threads. Also note that if **Robots robots.txt** is `Y`, then a `Crawl-delay` value in a site's `robots.txt` will override this setting.

4.4.11 Parallelism

Syntax: whole numbers from 1 up

Threads

This is the maximum number of simultaneous page fetching threads to allow against each site. Setting Threads higher than 5 is probably not very helpful, unless you have many "Single Pages" that are on various hosts.

Servers

This is the maximum number of different web servers to walk simultaneously. Setting this too high can stress your memory, cpu, and network.

4.4.12 Verbosity

Syntax: whole number from 0 through 4

Sets how much information the walker should provide about what it's doing. The default verbosity level is 2. The values are described in the following table.

Table 4.1: Verbosity Levels

Level	Description
0	Issue no messages except errors
1	Display starting point URLs
2	Display selected setting info
3	List URLs found in URL files
4	Indicate why URLs are rejected

The levels are cumulative. In other words, each level includes the previous levels.

Warning: at Verbosity 4, full Primer URLs will be printed to the Walk Status Log. If you use Primer URLs that contain credentials that you don't want other Webinator administrators to see, you will need to restrict access to the Walk Status, in addition to the Primer URL, when using Verbosity 4.

4.4.13 Disable Starting Walks

When set to `Y`, no walk will launch for this profile for any reason (manually run, schedule, etc). Note that even if set to `N`, walks may still be globally disabled if the System Wide Setting `Disable Starting All Walks` is set.

This can be useful with profiles that should be dataload-only, or for profiles that want to guarantee their content won't change.

Walks that are already running when this is set will finish normally.

4.4.14 Rewalk Type

Syntax: select from drop down box

This determines how rewalks are performed.

New

The default Rewalk Type, a `New` walk behaves just like the initial walk. The Webinator creates a new database and does a complete walk of everything, starting with the Base URLs. A `New` walk does not disturb the existing database during the walk.

- **When to use `New` walks** - `New` walks are useful when first setting up a profile and changing walk settings, as you're guaranteed to see your setting changes reflected in each document when a walk finishes. If you later make significant walk setting changes, it is recommended to do a `New` walk to make sure your indexed data reflects your new settings.

  Once your settings are established, though, fully processing every URL on every walk can be inefficient, and a different walk type may be more appropriate.

Refresh All

`Refresh All` walks start with the indexed data, and check each of the URLs to see if the content has changed. New URLs are added to the database, and URLs that are no longer present on the server are removed from the database.

If a URL's content hasn't changed, the Webinator doesn't reprocess the file. If the server supports `If-Modified-Since` (or it's doing a `file://` walk), the content won't even be transferred. This lets the walk be much more efficient.

- **When to use `Refresh All` walks** - `Refresh All` walks are useful for keeping content up to date once you've established all your walk settings. You're guaranteed for the walk to see anything that's changed, without needing to fully reprocess every URL every time.

  However, `Refresh All` walks don't apply the walk settings every walk. A new **Data from Field** rule to customize the Title will not take effect if a URL's contents hasn't changed. If you change your settings to include more URLs (i.e. add extensions, remove exclusions, add domains, etc.), a `Refresh All` walk is not likely to find the newly allowed data, unless all of the URLs leading to this data have been modified. You should do a `New` walk once to process these changes.

  For some large collections, especially those whose servers don't support `If-Modified-Since`, checking every URL every walk may still be too intensive. For these, `Refresh` walks can be used (see below).

  If more than 30%-50% of your site changes between walks you may be better off using a `New` walk instead of `Refresh All`. Also, many dynamic content generators may not give accurate `Last-Modified` dates, which will cause every URL to be rewalked. In that case you should use `New` instead of `Refresh All`.

Refresh

A `Refresh` walk behaves like a `Refresh All` walk, but it doesn't check every URL every walk. The Webinator pays attention to how often each URL changes, and schedules checking the URL less often if a URL isn't changed. When a `Refresh` walk starts, it only refreshes URLs that are scheduled for update at the start of the walk.

The idea is that if a profile is doing nightly walks and a URL hasn't changed in the last 6 months, it probably doesn't need checked EVERY night. It can be checked every 2nd night, every 3rd night, every 5th night, etc. as it continues to not change.

- **When to use** `Refresh` **walks** - `Refresh` walks are useful with a large (200k+ URL) collection of content that doesn't change very often, where the collection is too large to perform a `Refresh All` walk in a timely manner and dataload isn't possible. `Refresh` walks can finish much faster than a `Refresh All` walk. This allows another walk to start sooner and frequently-changing content to be re-checked sooner, instead of taking the time to finish refreshing all of the almost-never-changing content first.

  The downside of `Refresh` walks is that if a URL whose content rarely changes *does* change, it may not be picked up in the next walk because that URL may not be scheduled to be checked in the next walk. It may be worthwhile to schedule or manually launch an occasional `Refresh All` walk to check content slightly more often.

Singles Only

The `Singles Only` rewalk type is rarely needed, and only in specific scenarios.

`Singles Only` is like a refresh walk (doesn't create a new database), but it skips all the normal walking like `Base URLs` and refreshing content in the index. Instead it only walks "singles" settings (Single Pages, Single URLs, and Single Files). Further, every URL from singles is checked on every walk, regardless of whether it would be scheduled based on the refresh schedule described earlier.

- **When to use** `Singles Only` **walks** - `Singles Only` walks can be useful in scenarios where customers want something more efficient than refresh walking, like a dataload environment, but aren't able to construct proper dataload requests. If customers can produce a "changelist" URL that automatically lists all URLs that have changed recently, then that changelist URL can be named as a Single URL. A `Singles Only` walk will walk those URLs, without attempting to refresh the rest of the indexed content that the customer knows hasn't changed.

Rewalk Type Summary Table

The following table summarizes the trade-offs for the new and refresh rewalk types.

Method	Advantages	Disadvantages
`New`	Guarantees most accurate representation of current site. Does not disturb live search database.	Uses more bandwidth and temporary disk space. Longer time before site changes are reflected in live search.
`Refresh`, `Refresh All`, or `Singles Only`	Faster.	Could get out of sync with actual site under rare circumstances.
	Uses less bandwidth and temporary disk space. Site changes are reflected in live search much sooner.	A lot of changed pages could substantially slow searches during the walk. Works best with `If-Modified-Since` support on walked web server.

4.4.15 Rewalk Schedule

Syntax: select from drop down boxes

This performs a rewalk on the schedule specified. The rewalk action is the same as the one that can be started manually by clicking the `GO` button.

The `Frequency` defines how often to automatically rewalk.

The `Hour` defines which hour to start the rewalk for daily or weekly runs. You can click to select an hour from the drop-down list, or type in a more granular time (like `3:21 AM`).

The `Rewalk Type` defines what type of walk to perform. By default it uses the current `Rewalk Type` setting (see 4.4.14), but this allows a scheduled walk to override it.

You can define multiple walk schedules for the same profile by clicking the `Add More Schedules` link. This gives you more granular control in setting schedules. For example, instead of choosing between `once a day` and `once an hour`, you can have a walk launch 3 times a day by making the 3 schedules

- `Daily at 8:00 AM`
- `Daily at 12:00 PM`
- `Daily at 4:00 PM`

To remove a schedule, set its `Frequency` to `-None-` or click the red `X` to the left of the row.

See also `End of Walk Email` 4.5.2. If you are using "`On Change`" see also `Watch URL` 4.5.1.

4.4.16 Action Buttons

These buttons tell Webinator to do something now. Only the buttons applicable to the current status are displayed. The buttons are as follows:

- `Update`: Save the current settings for future use but don't begin a walk.
- `GO`: Begin a walk using the current settings.
- `Update and GO`: Save the current settings then begin a walk using those settings.
- `STOP`: Stop and abandon the walk that is currently running.

See the Walk Settings section (4.3) for details about the operation of these buttons.

4.5 Advanced Walk Settings

These are the advanced settings that are used less commonly than the settings available in `Basic Settings`. The advanced settings are available in `All Walk Settings`. You are not limited to the features listed here. You may modify the `dowalk` script to create additional features and to make the walk behave however you want it to behave.

See also `Customizing the Walker` 6.8 for information about the inner workings of the `dowalk` script.

4.5.1 Watch URL

Syntax: an HTTP URL

The URL specified here will be refreshed every time that Webinator starts a refresh walk. This can be used if you have a page that lists new documents that are added to the site as it will ensure that the links are found as soon as possible.

4.5.2 End of Walk Email

Syntax: an email address

If this is set, a summary report will be sent to the supplied email address when a walk occurs.

4.5.3 Attach Logs

This selects the log files to attach to the walk notification. The log files and walk errors are for the period of the refresh walk, and are sent as tab separated files that can be opened with programs such as Excel for further processing.

If the query log is attached it will be cleared after being emailed. This is an alternative to separate query log rotation and emailing and is particularly useful when using mode new for rewalks and you don't want to lose the query log. The query log is compressed for delivery with "zip" if present. If you want to use another program or your zip executable is not where dowalk expects you can modify dowalk and set `$zipexe` to the full path of your zip program. If your program uses different command line options than

zip you'll also need to adjust the <exec> lines where $zipexe is used to accommodate your program. If $zipexe doesn't exist the log will be emailed uncompressed so not having zip won't preclude receiving the logs, though they may be large and be rejected by some email systems due to size. See also Rotate Schedule (section 4.6.3).

4.5.4 Categories

Syntax: textual name and URL pattern pairs, additional input boxes will appear as you fill the ones provided

Webinator can create searchable sub-categories that will appear in a drop down box on the Search page. Enter the name of the category on the left, and its corresponding URL pattern on the right. URL patterns must fully match the URL (e.g. including protocol), and may contain asterisk (*) to indicate "anything" or question mark (?) to indicate any single character. There may be more than one pattern for each category; separate multiple patterns with space. Category names must not contain the pipe ("|") character, as it may be used to separate multiple categories in the category search parameter. A category should also not be named "Everything", as the search interface provides that option in the category selection box to search everything (i.e. any category), which might be confused with a specific category of the same name.

The following table provides an example.

Table 4.2: Example Categories

Category	URL Pattern
Demonstrations	http://www.example.com/demos/*
Manuals	http://www.example.com/manual/*
Books	http://www.example.com/a1/* http://example.com/b3/*

This example would create a category named Demonstrations which would only search the URL http://www.example.com/demos/ and any files under this directory, thereby creating a more concise match to the user's search. The same is true for Manuals. However, the Books category would include pages from both the /a1 and /b3 directories. The user would now have the option to search within just these categories or the entire database. The pattern should *not* be a single page unless you want a category with just that single page in it (e.g. http://www.example.com/manual/index.html or http://www.example.com/manual/ would generally be incorrect). It should typically be a prefix for a directory that has multiple pages within it, followed by an asterisk (*).

Note that **URL Pattern**s will not be used to determine categories if any **Data From Field** rules set Category. Please see the **Data from Field** settings (p. 62) for more details.

For best search performance, categories that overlap one another (i.e. contain walked pages in common) should be avoided if possible. If overlapping categories *are* used, they should be listed most-commonly-searched first. Also, the CatnoLowest field should be selected as one of the **Compound Index Fields** (p. 76); this is the default. These guidelines will allow the Auto-detect mode to optimize the most searches to the fastest possible speed.

Also note that changing, deleting or adding Category and/or URL Pattern *after* a walk has been performed will trigger a recategorization. This procedure, which runs in the background, re-applies the category changes to the walked data. While it is faster than a full walk – as pages do not need to be fetched

and fully processed – it nonetheless can take some time, particularly for large walks. For best performance, wait for the recategorization to complete (it can be monitored on the Dashboard or Walk Status as a task) before starting another walk.

4.5.5 Categories Type

Syntax: radio button choice

The **Categories Type** setting sets what type of categories are being used, and how to optimize category searches. It set to one of:

- `Auto-detect`
  Automatically detect what kind of categories are being used at search time, and optimize searches accordingly. This lets non-overlapping categories (i.e. those whose pages do not occur in any other category) be searched fastest, while still supporting overlapping categories as fast as possible. This is the default mode.
- `Overlapping`
  Assume that any category might overlap another. Category searches will be slower than with the other modes. This mode was used before the **Categories Type** setting existed. It can be set as a fallback if the cached overlap data is believed to be incorrect for some reason, e.g. category searches are wrong.
- `Non-overlapping`
  Assume that no category overlaps another. All category searches will be as fast as the fastest `Auto-detect` mode search, but searches for overlapping categories may not show all results. This mode can be set to force higher-performance searches at the potential expense of accuracy.

See the tips and performance caveats on the main **Categories** page (p. 55).

4.5.6 URL File

Syntax: the full path to a file on the web server's disk

This allows you to specify a file containing a list of site URLs to walk. This is an additional way of specifying more Base URLs 4.4.4. This file will be reread each time a rewalk is started. In the file, the list of URLs can be one URL per line (preferred) or delimited by any number of spaces.

4.5.7 URL URL

Syntax: an HTTP URL to a plain text file (*not* HTML)

This allows you to specify the URL of a plain text file containing a list of site URLs to walk. This is an additional way of specifying more Base URLs 4.4.4. This URL will be re-fetched each time a rewalk is started. In the file, the list of URLs can be one URL per line (preferred) or delimited by any number of spaces.

Warning: Due to the nature of `Stay Under`, a large number of URL URLs (1000+) in different directories will cause the walk to progress very slowly, as all URLs encountered will need to be checked against every one of those directories. In such a situation, we recommend turning off `Stay Under` and instead writing your own `Required Prefix`/`Required REX` expressions, which will be more efficient.

4.5.8 Single Page

Syntax: one or more URLs, one per line

Here you may specify URLs for individual pages to include in the index. These pages are fetched and stored in the database like others but the hyperlinks on them are not followed during a walk.

Pages removed from the list will *not* be removed from the database until the next `New` walk.

Note that since this setting is intended for "one-off" individual URLs, `robots.txt` will not be fetched for these pages.

4.5.9 Page File

Syntax: the full path to a file on the web server's disk

This may be used to specify a file containing URLs for individual pages.

Pages removed from the file will *not* be removed from the database until the next `New` walk.

In the file, the list of URLs can be one URL per line (preferred) or delimited by any number of spaces.

See also `Single page` 4.5.8.

4.5.10 Page URL

Syntax: an HTTP URL to a plain text file (*not* HTML)

This may be used to specify the URL for a plain text file containing URLs for individual pages. In the file, the list of URLs can be one URL per line (preferred) or delimited by any number of spaces.

Pages removed from the file will *not* be removed from the database until the next `New` walk.

See also `Single page` 4.5.8.

4.5.11 Strip Queries

Syntax: select Yes or No button

Strip query strings from all URLs. Some URLs have query strings on the end indicated by a question mark (`?`). With this option set to Yes, all query strings are removed from URLs before they are processed or retrieved.

4.5.12 Keep Query Vars

Syntax: comma-separated list of query variable names

If set, the Webinator will remove all but the these query variables when walking URLs. This can be useful when you know only a few select query variables are important, and others may appear but are irrelevant.

All the named variables are not required to be present, they are simply the ones that will be let through.

For example, setting Keep Query Vars to `id,type` will turn the URL

```
http://www.example.com//page.aspx?state=A3N4&id=432&printit=Y
```

into

```
http://www.example.com//page.aspx?id=432
```

4.5.13 Ignore Query Vars

Syntax: comma-separated list of query variable names

If set, the Webinator will remove these query variables when walking URLs. This can be useful when you can establish a few irrelevant query variables, but anything else would be significant.

URLs with these query variables do not cause any kind of error, they simply have the variables stripped out and continue processing normally.

For example, setting Ignore Query Vars to `printit` will turn the URL

```
http://www.example.com//page.aspx?state=A3N4&id=432&printit=Y
```

into

```
http://www.example.com//page.aspx?state=A3N4&id=432
```

4.5.14 Sort Query Vars

Syntax: select Yes or No button

This tells Webinator to sort the variable parameters in URL query strings.

Sometimes sites specify parameters in various orders but the returned content is the same. In such cases using this option can reduce the amount of time, bandwidth, and processing involved in downloading and processing those pages only to discard them as duplicates.

With this option on both of these URLs

```
http://www.example.com//page.aspx?state=A3N4&id=432&printit=Y
http://www.example.com//page.aspx?id=432&state=A3N4&printit=Y
```

will become

```
http://www.example.com//page.aspx?id=432&printit=Y&state=A3N4
```

4.5.15 Lower Query Var Values

Syntax: select Yes or No button

This tells Webinator to force the values (but not names) of variable parameters in URL query strings to lowercase.

Sometimes sites specify parameters with mixed capitalization but the returned content is the same. In such cases using this option can reduce the amount of time, bandwidth, and processing involved in downloading and processing those pages only to discard them as duplicates.

With this option on this URL

`http://www.example.com//page.aspx?id=432&state=A3N4&printit=Y`

will become

`http://www.example.com//page.aspx?id=432&state=a3n4&printit=y`

4.5.16 Ignore Case

Syntax: select Yes or No button

This tells Webinator whether to ignore case in URLs or not. The case of protocols and hostnames is always ignored but the case of paths and filenames is respected normally (if **Ignore Case** is `N`). Some web servers do not respect case, and the same file might thus be properly linked with differently-cased-paths URLs. In such cases **Ignore Case** `Y` will treat differently-cased-paths URLs as the same URL (but will preserve the case of the first variant found).

4.5.17 Extra Domains

Syntax: one or more domain names separated by space or line break

Allow walk to fetch pages from any host in the specified domain(s). Any URL (of any protocol) with a hostname ending in any of the specified domains will be accepted.

E.g. given **Base URLs** of `http://www.example.com/` and **Extra Domains** `othersite.com`, Webinator will walk all of `www.example.com`, as well as any URLs referring to any machine in `othersite.com` or its sub-domains (e.g. `docs.othersite.com`).

This option is not a "restricter" but an "enabler". All hosts specified will be walked and any others that match the given domain(s) will also be walked.

Note: This option does not *direct* the walk to web servers in the specified domains, like putting them in **Base URLs** would. It simply *allows* walking them – *if* a reference to them is encountered via walking the existing **Base URLs** etc. Thus if no links to **Extra Domains** are encountered, none will be walked.

4.5.18 Extra Networks

Syntax: one or more IP address prefixes separated by space or line break

Allow walk to fetch pages from any host within the network specified by the numeric IP address(es).

e.g.: Given a base URL of `http://www.example.com/` and extra network `192.0.2` Webinator will walk all of `www.example.com` and any URLs referring to any machine having an IP address prefix matching `192.0.2`.

Note: This option does NOT direct the walk to completely index every web server in the specified network. It simply allows walking them if a reference to them is encountered.

Note: Using this option has the potential to slow the walk, because every URL's hostname must be looked up. If there are many different off-site hosts, or your DNS is slow, the walk may be slowed substantially.

4.5.19 Extra URLs REX

Syntax: zero or more regular expressions (REX), separated by space or line break

Restricts walks to fetch URLs only matching any of the specified regular expressions anywhere in the URL (hostname, path, or query) when the Base URL matches.

If a Base URL is matched by an Extra URLs REX, then the only URLs that match the Extra URLs REX will be walked on that host. If a Base URL does not match an Extra URLs REX, then it is walked as normal.

It is a rarely used setting, most commonly used in conjunction with a hostname to fetch matching URLs on an additional host. Links still need to be found to those pages for them to be indexed.

For example, with the following Extra URLs REX:

```
>>=http://products\.example\.com=!supplierid+supplierid\=BigCo
```

(which matches a URL that begins with `products.example.com` and contains `supplierid=BigCo`), and using the following Base URLs:

```
http://products.example.com/listProducts.aspx?supplierid=BigCo
http://help.example.com/index.aspx
```

The Extra URLs REX matches the `products.example.com` URL, so only pages with `supplier=BigCo` will be walked, while all of `help.example.com` will be walked (following other inclusion/exclusion rules).

Available from version `4.3.9`.

See also `Extra Domains`, p. 59. See p. 193 for details on REX search syntax.

4.5.20 Exclusion REX

Syntax: zero or more regular expressions (REX), each on a separate line

Excludes URLs matching any of the specified regular expressions anywhere in the URL (hostname, path, or

query).

Table 4.3: Exclusion REX examples

REX	Matches
`/scratch[0-9]/`	a subdirectory named `scratch` followed by a single digit
`[^\alnum]test[^\alnum]`	the word `test` (but not `retest` or `tester` etc.)

See also `Exclusions` 4.4.9, `Exclusion prefix` 4.5.21 and `Exclude by Field` 4.5.23. See p. 193 for details on REX search syntax.

4.5.21 Exclusion Prefix

Syntax: zero or more URL prefixes, each on a separate line

Excludes URLs beginning with any of the specified prefixes. The entire URL (hostname, path, and query) is used for comparison.

Examples:

```
http://www.example.com/scratch0/
http://www.example.com/scratch1/
http://www.example.com/books/t
```

See also `Exclusions` 4.4.9, `Exclusion REX` 4.5.20 and `Exclude by Field` 4.5.23.

4.5.22 RSS Feeds

Syntax: select from options

`RSS Feeds` determines the behavior taken when a RSS or Atom feed is encountered during a walk, either by directly linking or an embedded `<link rel="alternate">`.

- `Follow Links Only` *(default)* - Links listed in the feed are followed, but the text content feed itself is not indexed.
- `Index Content and Follow Links` - Links listed in the feed are followed, and the feed itself is considered searchable content. The title and description of the feed are indexed, as well as the titles of the entries in the feed.

4.5.23 Exclude by Field

Syntax: Metamorph query, field to search, what to exclude

This provides more flexible control of what to exclude and how to exclude it. One exclusion per row of controls may be entered; new blank rows will be provided as rows are used. The `Metamorph Query`

column is where a Metamorph query (i.e. a typical search on Webinator) is entered: e.g. several keywords or a regular expression. The `Field` and `Meta Field` columns determine what the `Metamorph Query` searches: if `Meta Field` is non-blank, that named meta field is searched, otherwise the field selected in `Field` is searched. The `Exclude` column controls the action for pages that match the query: `Pages and links` indicates that both the matching page and its links are to be excluded; `Pages only` indicates that the matching page is to be excluded but its links are still followed – this is useful for excluding navigation-only pages; `Links only` indicates that the page is still included but its links are excluded.

See also `Exclusions` 4.4.9 and `Exclusion REX` 4.5.20.

4.5.24 Additional Fields

Syntax: Name, Type, Searchable, Sortable, Output

The additional fields allow you to add up to three additional fields to the index. These fields can be included in the output (if you use the XML output style), sorted on, and searched on. They are populated with the **Data from Field** settings (p. 62).

Additional Fields are supported in the full Texis product, but not Webinator-only.

- **Name** - specifies the name of the additional field. It also specifies element that will hold the field contents if it is output in XML. The name must be a valid XML element name (may contain only `alnum` or −_ and must start with a letter or _).
- **Type** - specifies the internal storage type for the additional field. Anything can be stored as `Text`, but if you want to do numeric or date comparisons (such as sorting), you have to use an appropriate data type.
- **Searchable** - specifies whether this additional field is directly searchable. This is done with an additional URL parameter that is separate from the normal query. Please see the **Additional Fields** section of **Procedures and Examples**, p. 176, for more details.
- **Sortable** - specifies whether you allow sorting by this additional field. This is done with the `order` search parameter. See the **Sorting** section of **Additional Fields**, p. 177, for more details.
- **Output** - specifies whether this field should be included with the output for XML results. Note that this ONLY refers to XML output, none of the "stock" results styles will include additional fields. If you want an additional field to show up in your search results, you must set **Output** to `Y` for the field, use the `XSL Stylesheet` **Results Style**, and customize the stylesheet to display the element for the Additional Field.

Note that once Additional Fields are created and used, changing their order or **Type** may alter other settings that use them, such as **Data from Field** (p. 62), **Index Fields** (p. 76) or **Compound Index Fields** (p. 76).

4.5.25 Data from Field

Syntax: REX expression, Replace expression, field to search, where to store it

This provides alternate means of setting both the HTML fields (`Modified`, `Title`, `Description` etc.) and any Additional Fields. It allows getting page information from non-default places by searching and optionally replacing the data. New blank rows will be provided as rows are used. See below for examples.

REX Search - Allows you to specify a REX expression to narrow down what contents of the `From Field` will be used. Leave it empty to use the entire field. See p. 193 for details on REX search syntax.

Note that a `REX Search` *must* be specified for the following `From` field types:

- `HTML`
- `HTML, raw output`
- `Text`

You can specify they entire field for these by using `.+` as the `REX Search`.

Replace - `Replace` can be used to specify a subset of the value to be stored in the `To` field (or subset of the match, if `REX Search` is used. See p. 198 for details on REX replace syntax.

From Field - specifies what the source field is for the data.

- `HTML` - the raw HTML source of the page. After matching, HTML tags are removed and HTML entities are resolved.
- `HTML, raw output` - the raw HTML source of the page. Content is left as-is, with tags in place.
- `Text` - the text of the page, after HTML rendering has been applied.
- `Title` - the HTML title of the page
- `All Meta` - the contents of all `meta` headers specified in the HTML page.
- `Meta Field ->` - the contents of a specific `meta` field, specified in the next input box, **From Meta Field**.
- `Keywords` - the contents of the `keywords` meta header.
- `Description` - the contents of the `description` meta header.
- `Mime Type` the MIME type of the page. This may have been derived from the `Content-Type` header, a `<META HTTP-EQUIV>` tag, or the URL extension, depending on what is available.
- `URL` - the URL of the page.
- `URL Decoded` - the decoded version of the URL. Any `%XX` 'URL-safe' sequences in the URL are replaced with their real characters. E.g. `Pre%20%2D%20Expense%20Report.doc` is decoded into `Pre - Expense Report.doc`.
- `URL Protocol` - the URL's protocol, e.g. `http`.
- `URL Host` - the host (without port number) from the URL.

- `URL Host and Port` - the host (and port number if given) from the URL.
- `URL Path` - the file path from the URL.
- `URL Path Decoded` - the file path from the URL, URL-decoded.
- `URL Anchor` - the anchor from the URL (if any), i.e. the part after the # (pound sign). May not be available if already stripped.
- `URL Query` - the query string from the URL (if any), i.e. the part after the ? (question mark).
- `URL Query Var ->` - the value of the URL query-string variable named in **From Meta Field**, URL-decoded.
- `Referrer's Data` - the value of a referring pages field. Store refs is required for this. The field selected will be the same field being populated.

From Meta Field - If `Meta Field ->` or `URL Query Var ->` is given as the **From Field**, this field is used to specify which meta field's or query var's contents to use as data. Leave blank otherwise.

Entering text in this field will force the use of `Meta Field ->`, if From Field is set to anything besides Meta Field or URL Query Var.

To Field - specifies where information should be stored.

- `Modified`, `Title`, `Description`, `Keywords`, `Depth`, and `Body` - Override the standard fields extracted from the content.
- `Authorization URL` - Populates the URL used when checking this result for Results Authorization. Please see the `Allow Authorization URL` section (4.6.55) for more details.
- `Category` - To populate the category via **Data From Field**, all the possible category names must be entered in the Category setting. Using one or more **Data From Field** rules to set Category will cause Webinator to ignore the Categories' URL Patterns and instead set category membership based on these **Data From Field** rules.

  Note: due to the way categories are stored, if categories are added, reordered, or removed after content has been walked, then a New walk will need to be performed to update the content's categories. Renaming categories does not need a rewalk.
- `Additional Links` - This target allows you to use **Data From Field** to create links that will be walked. These links are subject to the normal indexing rules, will be rejected if they match exclusions, etc.

  Use of this **Data From Field** target has no effect on the existing links found on the current URL. The links generated by this target will be added to the standard set of links on the page.
- `Subfetch` - This causes the Webinator to take the value(s) it finds and performs a fetch as URL(s). The URL can be absolute, or relative to the current URL.

  Nothing is changed by the subfetch itself, but any further **Data From Field** rules will use that fetched document(s) as the source of its content. Please see the Subfetch example below for a situation where this could be used.

- `Additional Fields` - If this profile has any Additional Fields, they will be available as a target `To Field`.

  If you just added the name of a new Additional Field, you will need to hit `Update` for the new Additional Field to appear in the `To Field` list.

  Additional Fields are supported in the full Texis product, but not Webinator-only.

Append - If set to `Y`, then the **Data From Field** content will be appended to the field's existing data instead of overwriting it. Date-type targets, such as `Modified`, do not support `Append`.

Data From Field Example - Using Description for Title

If there's a site that uses the same HTML title for every page but has a nice description, you can use the following settings to store the description in the `title` field (in addition to the `description` field).

- **REX Search**: *(Empty)*
- **Replace**: *(Empty)*
- **From Field**: `Description`
- **From Meta Field**: *(Empty)*
- **To Field**: `Title`

Data From Field Example - Using PublishDate for Last Modified Date

If you're walking a site of articles that specify a `PublishDate` meta field for every page, you can use that field's value instead of the normal `Last-Modified` date.

- **REX Search**: *(Empty)*
- **Replace**: *(Empty)*
- **From Field**: `Meta Field ->`
- **From Meta Field**: PublishDate
- **To Field**: `Modified`

Data From Field Example - Grabbing Price from Meta

If the site being walked defines a meta header on each page containing a price, it's possible to store that numeric data in an Additional Field for searching. Assuming you've already defined an Additional Field

called `Price`, the following settings would save that meta field in the Additional Field.

- **REX Search**: *(Empty)*
- **Replace**: *(Empty)*
- **From Field**: `Meta Field ->`
- **From Meta Field**: Price
- **To Field**: `Price`

Data From Field Example - Grabbing Price from Text

The target site might not be organized enough to stick the Price value in a meta header. If every page contains text in the format `Price: $19.95`, **Data From Field** can key in on that.

- **REX Search**: `Price:=\space+\$\P=[0-9\.]+`
- **Replace**: *(Empty)*
- **From Field**: `Text`
- **From Meta Field**: *(Empty)*
- **To Field**: `Price`

Notice that we use the field `Text` as the source, not `HTML`. By operating on the formatted text instead of the raw HTML source, it allows proper operation even if the HTML source uses things like `Price: <b>$19.95</b>` or `<td>Price:</td><td>$19.95</td>`.

Data From Field Example - Subfetch to use PDF Contents for a Web Page

`Subfetch`es allow you to use content from other URLs to populate the current URL's record. We may have a site about articles, where each article has a web page describing the article, and a link to a PDF of the actual article. We'd like searches that match article contents to take us to the web page, not the article PDF itself.

If the web page has a meta header called "`pdfLink`" with a URL to the article PDF, we can use the body of the PDF as a replacement for the web page's body with two **Data from Field** rules like this:

First **Data from Field** rule:

- **REX Search**: *(Empty)*
- **Replace**: *(Empty)*
- **From Field**: `Meta Field ->`
- **From Meta Field**: pdfLink
- **To Field**: `Subfetch`

Second **Data from Field** rule:

- **REX Search**: `.+`
- **Replace**: *(Empty)*
- **From Field**: `Text`
- **From Meta Field**: *(Empty)*
- **To Field**: `Body`

The `Subfetch` **Data from Field** rule fetches the URL specified in the `pdfLink` header. While this grabs the PDF, it doesn't change anything on its own. We then pull from the PDF's text output, and use that as the `Body` of the current web page.

4.5.26 Required REX

Syntax: zero or more REX expressions, each on a separate line

If specified, *all* URLs walked by Webinator must match at least one of these expressions. Opposite of `Exclusion REX`. See p. 193 for details on REX search syntax.

4.5.27 Required Prefix

Syntax: zero or more URL prefixes, separated by whitespace

If specified, *all* URLs walked by Webinator must match at least one of these prefixes.

4.5.28 Max Page Size

Syntax: a whole number from 1 up

Sets retrieved page size limit to the specified number of bytes. Pages larger than the limit will be truncated - not discarded.

Note: PDF files tend to be very large for the amount of text contained within them. Truncated PDF files are not processable due to their design. Make sure this setting is large enough to handle the largest PDF file you want to index.

4.5.29 Max Pages

Syntax: a whole number from -1 up

Limits the number of pages retrieved in a run to the specified number. Use `-1` for no limit.

4.5.30 Max Bytes

Syntax: a whole number from -1 up

Limits the number of bytes retrieved in a walk to the specified number. Use `-1` for no limit. The actual limit is rounded up to include the size of the last page so that it does not get truncated.

4.5.31 Max Depth

Syntax: a whole number from -1 up

Limits the depth of page retrieval to the specified number. Use `-1` for no limit. Depth is determined by counting how many links were traversed to reach a particular page. The base URLs are all at depth 0. URLs referred to by the base URL are depth 1, and so on.

4.5.32 Max URL Size

Syntax: an integer from 1 through 2033

Limits the size of URLs walked. URLs longer than this will be skipped. Should not exceed 2033. The default is 1024.

4.5.33 Max Requests

Syntax: an integer greater than 0

This gives the maximum number of server requests (page fetches) to make on a single server connection (i.e. Keep-Alive requests), if the server and protocol support multiple requests. Multiple requests per connection increases walk speed, and is needed for Windows/NTLM-protected pages. The default is `100`.

4.5.34 Max Connection Lifetime

Syntax: an integer greater than 0

This gives the maximum lifetime (in seconds) for a connection to a server. Multiple requests per connection may be made (if the server and protocol support it) until the connection is this old. The default is `600` (i.e. ten minutes).

4.5.35 Page Timeout

Syntax: a whole number from 1 up

Causes Webinator to timeout after the specified number of seconds during each page fetch. This includes the time to lookup the IP address of the host, make the connection to the server, and download a single page. A timeout does not cause the entire process to quit. That page is just skipped and considered unavailable.

4.5.36 Meta Tags

Syntax: zero or more meta tag names, each on a separate line

This option tells Webinator to look for the specified meta data in fetched documents and store it in the database. Then, this data is included in text searches. The meta tags "Description" and "Keywords" do not need to be specified here because they will be indexed by default. See below.

4.5.37 Standard Meta

Syntax: select Yes or No button

This option indicates whether to automatically extract the standard meta tags "Description" and "Keywords" from HTML documents. If "Yes", description and keywords meta data will be extracted and stored in their own fields within the database, unlike other meta data which will be collected and placed together into a single meta field in the database. These meta tags will be included in the search with a higher precedence than other meta tags.

4.5.38 All Meta

Syntax: select Yes or No button

Extract all `<meta name>` data from HTML documents, and all meta data from `anytotx` plugin-processed documents, and place into the meta field for searching. This eliminates the need to know the name of all possible meta tags, but it also opens the possibility of recording all manner of nonsensical meta data.

4.5.39 Storage Charset

Syntax: standard IANA character set (charset) name

This sets the charset for storing page text in the database during walks. Pages will be translated to this charset when inserted. If a page cannot be translated, it is stored and labeled with its source charset (if

known). If left empty (the default) it is `UTF-8`. This charset should be a superset of `US-ASCII` (same 7-bit sequences), and translatable by Webinator from all walked pages' source charsets.

Note that this is *not* necessarily the charset that search results will be displayed in: see Display Charset under Search Settings. This setting is the default value for Display Charset; see notes under Display Charset.

4.5.40 Source Default Charset

Syntax: a standard IANA character set (charset) name

If the source charset for a walked URL is not labeled and cannot be determined, assume it is this character set. Default is `ISO-8859-1`. This should only be changed if a large number of walk pages are in an unlabeled different charset, e.g. a Windows charset.

4.5.41 XML UTF-8

Syntax: select `Y` or `N` button

Whether to attempt to clean up UTF-8 data for XML output: remove sequences and characters that are invalid for XML. Should be `Y` if XML output (e.g. **Results Style** set to `XSL Stylesheet`) is used (and **Storage Charset** should be empty). This helps avoid browser errors with XML pages. *Note:* if XML output is *not* being used, this should be set to `N`, as certain characters that are HTML-safe but not XML-safe will be removed if enabled.

4.5.42 Keep HTML

Syntax: select Yes or No buttons

Specifies whether to include the named type of text in the database.

ALT text

`ALT` text from `IMG` or `AREA` tags.

<**STRIKE**>

Text between `<STRIKE>` and `</STRIKE>` tags.

<**DEL**>

Text between `<DEL>` and `</DEL>` tags.

<**FORM**>

Text of form elements, such as `<input>` tags, `<select>` boxes, and `<textarea>` elements.

4.5.43 Keep Links

Syntax: select Yes or No buttons

Specifies whether to follow the named type of links when walking.

Stylesheet

If `Y`, links from `<LINK HREF=... REL=stylesheet>` tags will be indexed as searchable content. Note that non-`stylesheet` `<LINK>` tags will still be followed regardless of this setting.

The default is `N`.

Script

If `Y`, Javascript links from `<script src=...>` tags will be indexed as searchable content.

The default is `N`.

<FORM>

If `Y`, links from `<FORM ACTION=...>` tags will be indexed as searchable content. Without the rest of the form properly filled out, such links can often produce nuisance error pages from database-driven sites.

The default is `N`.

4.5.44 Remove Common

Syntax: select Yes or No button

This causes common leading and trailing text from pages to be removed from the database. This is good for eliminating navigation menus and other static boilerplate text at the beginning and/or end of each page.

4.5.45 Ignore Tags

Syntax: one or more pairs of strings

All data between specified begin and end tag pairs will be stripped from the HTML before the text is extracted (i.e. links are unaffected). These are simple strings, not patterns nor REX expressions, and the case is ignored. This is useful for excluding boilerplate or otherwise unwanted portions of HTML documents. Tag pairs should not nest nor overlap in documents. Documents with no begin tag will be unaffected. Documents with no end tag after the last begin tag will still discard HTML from the last begin tag to end of document.

4.5.46 Keep Tags

Syntax: one or more pairs of strings

All data *not* between specified begin and end tag pairs will be stripped from the HTML before the text is extracted (i.e. links are unaffected). These are simple strings, not patterns nor REX expressions, and the case is ignored. This is useful for extracting prime interest areas of HTML pages without the surrounding boilerplate. Tag pairs should not nest nor overlap in documents. Documents with no begin tag will be unaffected. Documents with no end tag after the last begin tag will still keep HTML from the last begin tag to end of document.

4.5.47 Ignore Characters

Syntax: List of characters

List characters here which should be removed from the text and query. These can be punctuation that is optional. Examples are optional characters in part numbers, phone numbers, etc. Take care to avoid removing important characters, which you may want to delimit words. E.g. with the setting "`-@`", the text "`part 123-45@6`" would be stored (and searchable as) "`part 123456`" instead. Space is ignored in the setting value, and case is ignored when matching characters.

4.5.48 Plugin Split

A group of settings that control whether and how to split `anytotx` plugin output into multiple sub-URLs in the table. Non-text files, such as PDFs, that `anytotx` processes are often very large or composed of sub-files. The **Plugin Split** setting allows these files to be split up for finer-grain searching. Split files will cause more than one URL to be entered in the `html` table (and thus also in potential search results) for the original URL. Such subsequent URLs will have an anchor appended to distinguish them from each other; usually this is the sub-file name, but it may be generic e.g. "`#part5`" if there are no sub-files. *Note:* adjusting any of these settings can affect the ability of `Refresh`-type rewalks to complete successfully (`New` walks operate as usual). *Note:* **Data from Field** and other walk processing is not currently performed on **Plugin Split** URLs.

Depth

The **Depth** setting controls at what depth to split `anytotx` output. Each time a multi-file archive is unpacked by `anytotx`, the depth increases. (Note that the depth does not increase with any subdir(s) that may be created by each unpacking.) **Depth** 0 (the default) means split at the top level (i.e. do not split). **Depth** 1 would therefore insert each file of a ZIP file as a separate URL in the table. Files deeper than the **Depth** setting are left merged; e.g. another ZIP file contained within a ZIP file would have its files' text remain merged at **Depth** 1.

Bytes

The **Bytes** setting controls how many bytes each part will be after the file has been split. The default of 0 indicates do not split. This is useful for large monolithic files that have no detectable sub-file or page structure. If both **Pages** and **Bytes** are set, the first limit reached is used for each part.

AtPage

The **AtPage** setting controls whether to force the **Bytes**-controlled splitting to occur at a page boundary (a Ctrl-L). Checking this may make each part arbitrarily larger than the **Bytes** setting, because a part may extend to the next page break. With this setting unchecked, a part may be up to 50% larger than the **Bytes** setting, because the page-break check will only go that far over the limit.

Pages

The **Pages** setting controls how many pages to group in a part. The default of 0 does not split at all. If both **Pages** and **Bytes** are set, the first limit reached is used for each part. For example, setting **Pages** to 10 and **Bytes** to 100000 would break at 10 pages or 100KB, whichever comes first. This is useful to catch page-bounded documents like PDFs, and simultaneously avoid generating huge text for non-paged

documents.

4.5.49 Language Analysis

If **Enable** is set to `Y`, pages walked are processed through the Language Analysis Module (LAM), obtained and installed separately. This module helps support searching in languages such as Chinese, Japanese and Korean, where there is often no whitespace to delineate one "word" (logogram, or group of characters) from another, making searching difficult. The Language Analysis Module inserts spaces between words in the text of such pages, enabling ordinary non-wildcard searches to match better. At search time, users' queries are also passed through the module, so that they can match the processed pages' text.

Language

A two-letter ISO 639 language code "hint" for the LAM. If all or a majority of the walked data is a single language, entering that language's code here will help the LAM process data better. The default is empty (no hint). Added in Texis version 6.00.1294975881 20110113.

Preserve 7-bit

Whether to preserve the separation of all-7-bit tokens. Sometimes the LAM will separate alphanumeric tokens that are not language words, e.g. part numbers, causing search problems. Setting this to `Y` will attempt to preserve the separation (or lack thereof) of all-7-bit tokens in the walked text.

4.5.50 CJK Mode

Syntax: select Yes or No

`CJK Mode` modifies the walk and the search for better handling many Chinese, Japanese, and Korean queries.

At index time, multi-byte UTF-8 characters are indexed as individual words. At search time, multi-byte UTF-8 characters in the query are separated by spaces, and quotes surround the sequence to make it a phrase.

This allows the query to match where spacing may cause it to otherwise not match.

4.5.51 Unknown File Formats

Syntax: select Exclude or Include

Unknown File Formats controls how files in an unknown format (e.g. binary content not identified as PDF, Word, text, etc.) are handled. If set to `Exclude` (the default), such unknown formats' data will be ignored; this avoids bloating the walk database and query autocomplete dictionary with garbage binary content.

If set to `Include`, the data will be included. This might help find words in otherwise-unsearchable binary files, but is unlikely to succeed: since the file format is unrecognized, all that can be done is a simple `strings`-like scan for ASCII words in the file. If the file does not store words in an ASCII format, only garbage binary content will be returned.

Note that unlabeled plain-text files – i.e. those not identified by MIME type nor by file extension "`.txt`" – will generally be identified by a natural language scan (if running Texis 7.01 or later), and properly passed as-is. This setting only applies to files that fail that test, i.e. are unlikely to be plain text. Added in Appliance 9 / Webinator 7.

4.5.52 PDF Title Action

Syntax: select option

Controls how titles are handled for PDF files.

- `Automatic (default)` - Use the internal title if one is set, otherwise generate a title.
- `Always Generate` - Always generate a title for the PDF, ignoring any title set in the document.
- `Never Generate` - Always use the internal title from the PDF. If there isn't one, the title is left empty.

Binary files like Word documents can have titles set internally. They're usually unset, and if they are set, it's likely to a good title that should be used.

PDF files often differ from this. Many PDF converters set the PDF's title to the original filename that the PDF came from. This results in a PDF with a title like `expenseReport.doc`, which shows up as the link text in the search result. Users click on what appears to be a Word document, only to arrive at a PDF document.

Setting `PDF Title Action` to `Always Generate` tells the Webinator to ignore any internally set title for PDF files, and always generate one based on the content.

4.5.53 Word Definition

Syntax: one or more regular expressions (REX), each on a separate line

Sets the word matching expression(s). Each line is a regular expression defining what is considered a word within the textual content of the retrieved documents during the index process. The default expressions index normal words and some special items such as domain names.

You may supply multiple expressions, one per line, if you can't define your idea of all possible words in one expression.

For example, `>>\alpha=\alnum{1,20}` will index "words" beginning with an alphabetic character followed by 1 to 20 alphabetic or numeric characters.

If **Word Definition** is changed, the **Language Characters** setting (p. 122) should generally be updated to reflect any new characters added.

Changing the word definition with `Update` instead of `Update and GO` will cause the existing search index on the data to be dropped and rebuilt. The database will not be searchable during the time that the index is being rebuilt; this may take several minutes or more for large profiles.

See p. 193 for details on REX search syntax.

4.5.54 Text Search Mode

Syntax: select from options or enter custom mode

(Note: In earlier releases this setting was known as **Character Match Mode**.)

Sets the character-matching mode for text (keyword) searches. This controls aspects like case-sensitivity, ignoring accents, etc. The selectable values are:

- `Loose` - Ignore case, ignore diacritics (accents), expand ligatures, ignore width differences. **Storage Charset** should be empty or `UTF-8`, though `ISO-8859-1` may sometimes work. With this mode, not only will a lower-case "`e`" match an upper-case "`E`" and vice-versa (ignore case), but "`e`" will match "é" (Unicode U+00E9), "`oe`" will match "œ" (U+0153), and full-width will match half-width characters (for ASCII and katakana).
- `Strict` - Ignore case only. "`e`" will match "`E`", but not "é". **Storage Charset** should be empty or `UTF-8`, though `ISO-8859-1` may sometimes work.
- `Strict ISO-8859-1` - Ignore case only, and assume **Storage Charset** is `ISO-8859-1`. For back-compatibility. Available only for **Text Search Mode**.
- `Exact` - Match characters exactly, respecting case, diacritics, width etc. Available only for **Attribute Compare Mode**.
- `Custom ->` - Use the custom mode entered in the **Custom Mode** box. This is a comma-separated list composed from the following tokens; consult Thunderstone tech support for advice:
  - `iso-8859-1` - Assume text is ISO-8859-1 encoded. Should only be used if **Storage Charset** is also `ISO-8859-1`. If this flag is not set, text is assumed to be UTF-8, though occasional ISO-8859-1 characters will usually be able to match their UTF-8 equivalents.
  - `ignorediacritics` - Ignore diacritic marks (accents, umlauts, etc.). E.g. "`e`" will match "é" (U+00E9) and vice-versa.
  - `expandligatures` - Expand ligature characters. E.g. "`oe`" will match "œ" (U+0153) and vice-versa. Note that with this flag off, certain ligatures may still be expanded if necessary for case-folding with `ignorecase`.
  - `ignorewidth` - Ignore half-/full-width differences, e.g. for ASCII and katakana characters.
  - `ignorecase` - Ignore case differences, e.g. "`e`" matches "`E`" and vice-versa; this is the default. The alternative is `respectcase`.
  - `respectcase` - Case-sensitive search, e.g. "`e`" does *not* match "`E`". The alternative is `ignorecase`.
  - `unicodemulti` - Use Unicode case-compare tables, with multi-character expansions where needed (e.g. for ligatures). The alternative is `ctype` or `unicodemono`.
  - `unicodemono` - Use Unicode case-compare tables, but do not expand characters. The alternative is `ctype` or `unicodemulti`.

- `ctype` - Use the operating system's `ctype.h` case-compare tables. Only codepoints U+0001 through U+00FF (i.e. single-byte or ISO-8859-1 range) are supported, though the actual encoding may be ISO-8859-1 or UTF-8 depending on the `iso-8859-1` flag. The alternative is `unicodemulti` or `unicodemono`.

Note: Changing the **Text Search Mode** setting will cause text search indexes to be rebuilt, which may take several minutes or more for large profiles.

4.5.55 Attribute Compare Mode

Syntax: select from options or enter custom mode

Sets the character-matching mode for attribute comparison searches, e.g. equals, less-than, order-by, `IN`. This controls aspects like case-sensitivity, ignoring accents, etc. See **Text Search Mode** (p. 75) for details on what the setting values mean. The default is `Exact`. Note that searches on `Enum` fields are unaffected by this setting, as the `Enum` type is defined to be case-insensitive.

Note: Changing the **Attribute Compare Mode** setting will cause **Extra Indexes** (if any) to be rebuilt. This may take a few minutes on large profiles, and may prevent walks from proceeding until the indexes finish.

4.5.56 Index Fields

Syntax: list of fields ordered by desired weight

These fields will be searched by the user's text query. Fields listed higher will be weighted higher in search results, according to the **Position in Text** search setting. **Additional Fields** may be selected, if they are to be searched by the user's text query. Note that changing **Additional Fields**' names, types or order later may affect their presence in **Index Fields**.

Note that changing these fields will cause indexes to be rebuilt, which may take several minutes or more for large-data profiles. The old setting will be used until the index rebuild is complete.

4.5.57 Compound Index Fields

Syntax: list of field(s) from select boxes, any order

These fields will be indexed along with **Index Fields**, but in the compound portion of the main search index. They are not searched by the text query, but are used to improve accuracy and performance for certain ancillary queries performed in *addition* to the main text search, such as when ordering results by date, or searching by depth. The default values are `Visited`, `Modified`, `Depth` and `Pop`.

The selected fields may be in any order; they are used only when needed, unlike **Index Fields**, all of which are always searched by the user's text query. Note the following caveats:

> Adding a field to **Compound Index Fields** will not help search performance if there is no main (text) query also, as the compound part of the index can only be used in conjunction with a text query.

Only a fixed-size amount of data can be stored in each row of each of the **Compound Index Fields**, so only fixed-size fields such as dates, integers, numbers, etc. should be chosen. If text data is used, all values for the field in the database should be small (a few characters) for best performance.

Note that as this is the same overall index as **Index Fields**, changing any of these fields will cause indexes to be rebuilt, which may take several minutes or more for large-data profiles. The old setting(s) will be used until the index rebuild is complete.

4.5.58 Extra Indexes

Syntax: select-box for index type and table, text box to enter index name and fields

Extra Indexes may be created to improve search performance and accuracy in situations where the main text index (**Index Fields**) and/or its **Compound Index Fields** are not sufficient. They are not generally created unless suggested by Thunderstone tech support for certain queries.

Note that creating an extra index on a large-data profile may take several minutes or more. If the index **Type** is not `Metamorph` nor `Metamorph Inverted`, creating the index may also impede walks or other database modifications. Non-`Metamorph`/`Metamorph Inverted` indexes should therefore be created *before* the profile is walked or populated with data to avoid this issue, if possible. **Extra Indexes** should only be created when the profile is not actively walking, to minimize load and potential walk impediments.

4.5.59 Spell-check Dictionaries

Syntax: select-box choice

This setting controls what dictionaries to create for spell checking. The default (`Create all`) is to create all needed dictionaries. However, this can consume significant time and memory for some large-data profiles, so to conserve system resources, only the multi-word-occurrence dictionary may be created (`Create multi-word only`). This may reduce spell-check suggestions at search time however. To further conserve system resources, no dictionaries at all may be created (`None`). This will disable spell checking at search time.

4.5.60 Primer Type

"Primer URLs" are URLs that are fetched before actually starting a walk. They are not stored in the search database, but instead are used to "prime" Webinator with any necessary credentials (e.g. login cookies) for accessing the rest of the site. By default, the Base URL is used, in case any session/ASP cookies are needed.

The **Primer Type** setting specifies which (if any) URLs are used to prime the profile:

- `None` - No primer URL is used. The Base URLs are walked as normal.
- `Base URL` - the Base URLs are used to prime the walk. This differs from `None` in that the base URLs are submitted once and the results discarded, and then submitted again for walking.

This is useful in situations where the **Base URL** contains login information, and the page returns "thank you for logging in" with no other content until the page is requested again.

- `Custom` *(default)* - The URLs listed in **Custom Primer URLs** (if any) are used, as described below.

For directly-supported authentication schemes – HTTP Basic, NTLM, Negotiate, CAS, SAML/ADFS, or file authentication – the **Login Info** setting should be used instead.

4.5.61 Primer URLs

Syntax: URL, optional variables, optional bad-login query, optional URL query

When the **Primer Type** setting is set to `Custom`, the **Primer URLs** setting values take effect. There are two ways to use a custom primer URL - submitting the form directly, and filling out the form.

Submitting the Form Directly: Custom Primer URL

If a form-based login must be filled out before accessing a site, the **Custom Primer URL** can be set to the `<FORM ACTION>` URL of the login (fully-qualified), with any form variables (e.g. user/pass) filled out in the query string. If the `<FORM METHOD>` must be `POST` instead of `GET`, the URL protocol may be changed to the pseudo-protocol "`http-post`". E.g.:

```
http-post://login.acme.com/checkLogin.asp?User=Admin&Pass=open-sesame
```

would be submitted using the `POST` method, with the given query-string variables sent as the content. Note that the query-string variables and values should be URL-encoded.

Filling Out the Form: Custom Primer Variables

Sometimes submitting the form directly is not sufficient. Forms on web pages can contain dynamic hidden variables, such as a `viewstate` for session tracking. This means the form must be opened, filled out, and submitted, instead of simply submitting a pre-defined action URL.

This is achievable with the **Custom Primer Variables** setting. Instead of setting **Custom Primer URL** to the action of the login form, you set it to the URL of the page that contains the form. **Custom Primer Variables** is a URL-encoded list of name/value pairs to set on the **Custom Primer URL** page.

When **Custom Primer Variables** is set, the **Custom Primer URL** is fetched, and then the variables specified in **Custom Primer Variables** are used on the form, and then *that* form is submitted.

For example, let's say there's a `pleaseLogin.asp` page that submits to `checkLogin.asp`, and the form contains a dynamic state that has to be included or `checkLogin.asp` will reject the login. If you set **Custom Primer URL** to

```
http://login.acme.com/pleaseLogin.asp
```

and set **Custom Primer Variables** to

```
User=Admin&Pass=open%26close
```

The `pleaseLogin.asp` page will be fetched, the form field `User` will be set to `Admin` and `Pass` will be set to `open&close` (note the URL-encoding), and then the form on the `pleaseLogin.asp` page will be submitted, going to `checkLogin.asp`.

This means that if the form on `pleaseLogin.asp` contains

```
<input type="hidden" name="sessionstate" value="abc123xyz"/>
```

then that hidden variable will be submitted along with the rest of the form.

Note: After version 6.1.4 (2012-07-13) Webinator will set the HTTP(S) `Referer` header for each primer URL to the URL of the previously used primer. So authentication systems that require `Referer`s will work. If the first primer URL also requires a `Referer` add a primer URL before that so it picks up that as the `Referer`. This does not affect the use of `Referer` in the main walk.

Checking for Bad Logins: Bad Login MM Query

Sometimes, the primer URL login may fail, e.g. bad login. However, since the only error indication may be a "Login failure"-type message and not a true HTTP error code, Webinator may not be able to detect this and might continue walking useless (permission-denied or "Please log in first") pages.

To help detect such a primer URL failure, a **Bad Login MM Query** may be entered. If non-empty, this is a Metamorph query to run against the HTML returned from the associated primer URL. If it matches, the primer URL is considered a failure, and the walk is stopped for that particular site (other Base URLs will continue).

Multiple Primers: Base URL MM Query

If multiple custom primer URLs are being used, you can control which ones are used for which Base URLs via Base URL MM Query.

By default, primer URLs are only used on Base URLs that have a matching protocol and hostname. If **Base URL MM Query** is non-empty, then this Metamorph query will be run against the Base URL being walked. The associated primer URL will only be fetched if it matches.

Following additional links with the !FOLLOW_LINK token

The primer system automatically follows the HTTP redirects `301 Moved Permanently` and `302 Found`. Sometimes login systems produce additional links that must be followed to get the login cookie, but aren't true HTTP redirects. Examples could be JavaScript that sets `document.location` or a page that simply says "click here to continue".

The special Custom Primer URL token `!FOLLOW_LINK` can be used instead of a URL to follow the first link generated by the previous primer's fetch. This can be added multiple times to follow multiple links.

4.5.62 Unprimer URLs

Syntax: URL, optional variables, optional bad-login query, optional URL query

"Unprimer URLs" are URLs that are fetched before finishing a walk on a site. They mirror the `Primer URL` (p. 78) settings and can be used for sites that require a logout, or otherwise should be notified the walk has finished.

Submitting the Form Directly: Custom Unprimer URL

If a form-based logout should be filled out before leaving a site, the **Custom Unprimer URL** can be set to the `<FORM ACTION>` URL of the login (fully-qualified), with any form variables (e.g. user/pass) filled out in the query string. If the `<FORM METHOD>` must be `POST` instead of `GET`, the URL protocol may be changed to the pseudo-protocol "`http-post`". E.g.:

```
http-post://login.acme.com/Logout.asp?User=Admin
```

would be submitted using the `POST` method, with the given query-string variables sent as the content. Note that the query-string variables and values should be URL-encoded.

Filling Out the Form: Custom Unprimer Variables

Sometimes submitting the form directly is not sufficient. Forms on web pages can contain dynamic hidden variables, such as a `viewstate` for session tracking. This means the form must be opened, filled out, and submitted, instead of simply submitting a pre-defined action URL.

This is achievable with the **Custom Unprimer Variables** setting. Instead of setting **Custom Unprimer URL** to the action of the form, you set it to the URL of the page that contains the form. **Custom Unprimer Variables** is a URL-encoded list of name/value pairs to set on the **Custom Unprimer URL** page.

When **Custom Unprimer Variables** is set, the **Custom Unprimer URL** is fetched, and then the variables specified in **Custom Unprimer Variables** are used on the form, and then *that* form is submitted.

Webinator will set the HTTP(S) `Referer` header for each unprimer URL to the URL of the previously URL so systems that require `Referer`s will work. If the first unprimer URL also requires a `Referer` add an unprimer URL before that so it picks up that as the `Referer`. This does not affect the use of `Referer` in the main walk.

Checking for Bad Logins: Bad Login MM Query

Sometimes, the unprimer URL login may fail, e.g. bad login. However, since the only error indication may be a "Login failure"-type message and not a true HTTP error code, Webinator may not be able to detect this

and might continue walking useless (permission-denied or "Please log in first") pages.

To help detect such a unprimer URL failure, a **Bad Login MM Query** may be entered. If non-empty, this is a Metamorph query to run against the HTML returned from the associated unprimer URL. If it matches, the unprimer URL is considered a failure, and the walk is stopped for that particular site (other Base URLs will continue).

Multiple Unprimers: Base URL MM Query

If multiple custom unprimer URLs are being used, you can control which ones are used for which Base URLs via Base URL MM Query.

By default, unprimer URLs are only used on Base URLs that have a matching protocol and hostname. If **Base URL MM Query** is non-empty, then this Metamorph query will be run against the Base URL being walked. The associated unprimer URL will only be fetched if it matches.

Following additional links with the !FOLLOW_LINK token

The primer system automatically follows the HTTP redirects `301 Moved Permanently` and `302 Found`. Sometimes login systems produce additional links that must be followed to get the login cookie, but aren't true HTTP redirects. Examples could be JavaScript that sets `document.location` or a page that simply says "click here to continue".

The special Custom Unprimer URL token `!FOLLOW_LINK` can be used instead of a URL to follow the first link generated by the previous primer's fetch. This can be added multiple times to follow multiple links.

4.5.63 Login Info

Syntax: name and password

Specify a username and password for sites that require a login to view certain pages. These are used with HTTP Basic, file, Windows NTLM, Negotiate, CAS, SAML/ADFS, and FTP authentication. Other authentication schemes are not supported currently, though many web-based schemes – e.g. a login form – may be accessible with a custom **Primer URL** (p. 78). Without proper login, protected pages will either result in a error and be skipped, or a "Please login"-type message will show on the walked page.

If this is a Windows domain account, enter both the domain and user name in the Username field, separated by a backslash (`\`), i.e. `MY_DOMAIN\myuser`.

Note: The search interface displays hit context and has an option to view the entire text of the page. This allows search users to view "protected" pages without entering a password.

4.5.64 Proxy Auto-Config URL

Syntax: the full URL to a proxy auto-config (PAC) script

This specifies the URL to a PAC script. The script is fetched once at the start of a walk, and then run for each URL walked, to determine the proxy (or direct fetch) to use for that URL. This setting overrides **Proxy**.

A proxy auto-config script can be used to dynamically configure the proxy to use on a URL-by-URL basis, instead of using one proxy for all URLs. The script can also return multiple proxies to use; e.g. a primary, and fallback(s) to use if the primary is unreachable. See the website `findproxyforurl.com` for more information on PAC scripts.

In Windows Control Panel, the `Internet Options` → `Connections` tab → `LAN settings` → `Use automatic configuration script` address value is equivalent to Webinator's **Proxy Auto-Config URL** setting.

4.5.65 Proxy

Syntax: the full URL to a web proxy server

This specifies the URL (not just hostname) of a proxy web server through which to pass page fetch requests. Only host and (optional) port are used from the given URL. If empty, no proxy is used: pages are fetched directly.

Note that **Proxy Auto-Config URL** overrides this.

In Windows Control Panel, the `Internet Options` → `Connections` tab → `LAN settings` → `Use a proxy server for your LAN` host/port value is equivalent to Webinator's **Proxy** setting.

4.5.66 Proxy Login Info

Sets the user name and password to authenticate to proxy servers, using the `Proxy-Authenticate` header and Basic Authentication. Used if the Proxy URL is filled in. Added in version 4.01.1031600000 Sep 9 2002.

4.5.67 Client Certificate

Chooses which client certificate (if any) to use when authenticating with HTTPS servers. These are normally not needed unless the remote server requires a client certificate for authorization.

Certificates are managed in the `Client Certificates` section (4.3).

4.5.68 Cookie Source Path

File path to a Netscape or Microsoft Internet Explorer format cookie file to read at start up. This allows persistent cookies saved by a browser to be read by Webinator, so it can inherit the browser's state. To easily walk a site that requires a custom login (i.e. not HTTP Basic authentication), and that uses persistent cookies, just login normally using a browser run *on* the Webinator machine itself. Then, enter that browser's cookie file in the Cookie Source Path setting (this is typically `%USERPROFILE%\Cookies` for Explorer

on Windows). Then, Webinator will automatically inherit the browser's permissions. Added in version 4.02.1042043803 Jan 8 2003.

4.5.69 Cookie Jar

Netscape or Microsoft Internet Explorer format cookie data to load at start up. An inline alternative to `Cookie Source Path`. This allows persistent cookies saved by a browser to be read by Webinator, so it can inherit the browser's state. To easily walk a site that requires a custom login (i.e. not HTTP Basic authentication), and that uses persistent cookies, just login normally using a browser. Then, copy that browser's cookie data into the `Cookie Jar` setting. Then, Webinator will automatically inherit the browser's permissions. Added after version 6.1.4 2012-07-13.

The Netscape format is one cookie per line, with the tab-separated values: `Domain IsOkAllDomain Path IsSecure IsHttpOnly Expires Name Value`. E.g. the line:

```
.example.com TRUE / FALSE FALSE 0 MyCookie MyValue
```

would represent a session cookie named `MyCookie` with value `MyValue` sent for any path for any site in the domain `.example.com`.

4.5.70 Strict Cookie Paths

If set to `Y`, the `Path` attribute (if present) of any received cookie must be a prefix of the URL setting it, or the cookie will be discarded, as per RFC 2965 3.3.2. This helps prevent one application from altering the cookie(s) of another application on the same server; such isolation may be desired if the applications should be protected from each other. However, typically such cross-path altering is acceptable – e.g. some login systems depend on it – so this setting defaults to `N`, which also aligns with typical browser behavior. Only available with products using Texis version 6.00.1342215000 201210713 and later; earlier versions effectively behave as if this setting were always `Y`.

4.5.71 Off-Site Pages

Syntax: select Yes or No button

Allow retrieval of individual off-site pages. By default Webinator will not retrieve pages that are not on the same host as the base URL(s). Using this option, pages not on the same machine will be retrieved, but none of the pages that they reference will be walked.

All other discarding rules don't involve the site except for Stay Under still apply (Extensions, Exclude Prefix, Required Rex, etc).

4.5.72 Off-Site Components

Syntax: select Yes or No button

This option also allows off-site resources embedded within on-site pages to be fetched for processing. This includes JavaScript sources, embedded frames, and redirects.

4.5.73 Stay Under

Syntax: select Yes or No button

When this flag is Yes, walks will stay under the directory specified in the base URL(s). When this is No, if a hyperlink to another location on the same site is encountered, the will follow the link. In neither case will the walk go to other sites unless they are in the list of walk URLs or allowed domains or networks.

4.5.74 Prevent Duplicates

Syntax: select Yes or No button

This option enables extra checking for duplicate documents. Documents with the same content are only be stored once, even if their URLs are different. This is accomplished by hashing the textual content of the page and not storing any page with a hash code that is already in the database.

4.5.75 Respect Canonical URLs

Syntax: select Yes or No button

This option enables respecting when pages indicate that there's a canonical version of themselves, expressed with `<link rel="canonical">`.

If `Y` (the default), then if a page has a `<link rel="canonical">` that is different from the fetched URL, the original page will NOT be indexed and the canonical URL will instead be added to the index.

Note that if the canonical URL is not be allowed due to walking rules (matches exclusions, off-site, etc), then the canonical behavior does not apply and the original, non-canonical URL will be indexed.

4.5.76 Duplicate Check Fields

Syntax: checkboxes to choose fields

These are the fields which will be checked for duplicate prevention (if `Prevent Duplicates` is enabled). The concatenation of these fields is hashed for each incoming document, and if the hash is the same as an existing document, the incoming document will be discarded as a duplicate.

By default, all fields are included, so any differences in the content of two documents will cause them to not be seen as duplicates.

Note: Changing `Duplicate Check Fields` after a walk has completed (i.e. before a later `Refresh` type walk) may cause new documents to not be removed as duplicates as expected, since the pre-existing documents' hashes are now for a different set of fields. This will not cause errors or corruption; it just might leave some newly-duplicate documents in the database.

4.5.77 Store Refs

Syntax: select Yes or No button

Controls whether URLs referenced by retrieved pages are added to the refs table. This can save some time during the walk, as well as, disk space if it's turned off. But turning it off prevents the "`Show Parents`" option in the search from working. It also reduces the detail available from walk error reports.

4.5.78 Inline Iframes

Syntax: select Yes or No button

This indicates whether to treat iframes as a part of the page they are on or as separate stand alone pages. Selecting Yes will make them part of the page. Selecting no will make them separate.

4.5.79 Max Components

Syntax: a whole number from 0 up

This indicates the maximum number of page components (frames, iframes, and JavaScript src) to fetch while processing a page. Pages with more components than this are discarded. If this is set to 0, the frames of framed documents are treated as independent, stand-alone pages.

4.5.80 Execute JavaScript

Syntax: select Yes or No button

Execute JavaScript that is contained on fetched pages and that might alter or generate the page content and URLs.

Note that the JavaScript engine in Webinator has limited functionality. Additionally, while a browser need only maintain the *current* JavaScript/DOM state of a page, a crawler ideally needs to know *all possible* states – often triggered by events from a user that does not exist – in order to find all dynamic links on a page; this may not be feasible. Thus, even with JavaScript enabled, many JavaScript-derived links and/or content may still not be found.

4.5.81 Fetch JavaScript

Syntax: select Yes or No button

Fetch JavaScript that resides at a separate URL instead of being inline on the page (e.g. `<SCRIPT SRC>` tags).

4.5.82 JavaScript String Links

Syntax: select appropriate checkboxes

Sets which additional sources of potential JavaScript links to check. Some JavaScript links may not be found when scripts on a walked page are executed, so the internal list of all JavaScript string objects is scanned for potential URLs according to the checked boxes. `Menu` will look for common JavaScript menu navigation system links; `Protocol` will look for strings that look like valid fully-qualified Web links; `File` will look for probable file strings.

Note that any of these sources may potentially find incorrect links, especially the `File` type. Checking `File` is generally used only as a last-ditch effort to find some JavaScript links.

4.5.83 Debug JavaScript

Syntax: select Yes or No button

Print additional debugging messages for JavaScript errors.

4.5.84 JavaScript Memory

Syntax: numeric memory size e.g. `20MB`

Alters the max amount of memory allowed for running JavaScript. The default (if the setting is empty) is 20MB. Increasing the limit may help if error messages such as "`JavaScript exceeded scriptmem limit`" are encountered. Note that the Maximum Process Size limit setting may also need to be increased if this is increased.

4.5.85 JavaScript Timeout

Syntax: integer

Max time, in seconds, to allow for running JavaScript. The default (if the setting is empty) is 5 seconds. Large or complex JavaScript pages may require more time, e.g. if "`JavaScript exceeded scripttimeout`" messages are received.

4.5.86 AJAX Crawlable URLs

Syntax: select Yes or No button

When enabled (the default), support the Google AJAX crawling scheme. This allows AJAX URLs (which are usually not walkable) to be walked, if the site being walked also supports the scheme.

AJAX URLs which contain anchors/fragments (`#someFragment`) are not normally walkable because anchors are never sent in HTTP requests, and the client-side JavaScript support in Webinator does not

include AJAX so the anchor is not processed by the walker either. Thus AJAX anchor links look like duplicates and are never fetched, or if fetched, do not return the anchor-specified content.

With the Google AJAX crawling scheme, walkers temporarily rewrite certain links with conforming anchors – those that begin with an exclamation point – by placing the anchor into the query string. Since query strings *are* sent in HTTP requests, the server sees the anchor, and can return the appropriate content. Moreover, the returned content can be static so that the walker can index it.

Specifically, URLs of the form:

```
http://example.com/path#!fragment
http://example.com/path?query=present#!fragment
```

will be requested from the server as (respectively):

```
http://example.com/path?_escaped_fragment_=fragment
http://example.com/path?query=present&_escaped_fragment_=fragment
```

This temporary rewrite is only used for the walker fetch: search results still return the original AJAX-anchor version of the link, so that browsers can still take advantage of the AJAX features of the site.

Note that this scheme requires web site support: the site must respond to any URL with the `_escaped_fragment_` query parameter with the appropriate, full, static HTML content that corresponds to the AJAX anchor state of the same value.

4.5.87 Walk Trace Settings

Syntax: text

Debug/trace settings and values for walks, as a comma-separated list of "`param=value`" tuples. These are generally only set at the request of Thunderstone tech support, as they can cause copious tracing messages to appear as walk "errors", some of which may reveal authentication or other sensitive information. Supported settings and values are subject to possible change in future releases. See also **Search Trace Settings**, p. 121, which has the same syntax.

4.5.88 Audit Log

Syntax: Y or N

If enabled, all setting changes for this profile are written to the `logs/audit.log` file in the installation directory. This includes where the event came from, which account did it, which profile, and what changed.

Note: While known sensitive fields (e.g. **Login Info**) have values redacted, other sensitive data may nonetheless be logged (e.g. URLs). See also the system-wide setting (section **Audit Logging** 4.7.7, p. 129).

4.5.89 Performance Logging

Syntax: Y or N

If enabled, a detailed log will be made for each walking process that catalogs the time taken in various steps of walking.

When viewing the details of a walk task (by clicking on its PID in Walk Status), you will see a link to `View Performance Data`. Clicking that link will display a graph of how long the process spent on each step of the walking process.

You can also view a profile's performance logs by clicking `Archived Logs` from the Walk Status page.

4.5.90 Batch Locks

Syntax: Y or N

If enabled, Webinator will use a more efficient method of locking tables during the indexing process.

There should be no reason not to do this during normal operation, and should only be disabled at the request of Thunderstone Support when troubleshooting other problems.

4.5.91 URL Protocols

Select which URL protocols to allow to be fetched. If a protocol is not enabled, but the **Base URL** uses it, it will be automatically enabled for the walk. The URL protocols supported are `http`, `https`, `ftp` and `gopher`.

4.5.92 HTTP Version

What HTTP version to use for requests. `HTTP/1.1` enables compression (`gzip`, `chunked`, `compress`, `deflate` `Content-Encoding`) and is the default for products using Texis version 6 and later. `HTTP/1.0` was the default for previous versions. `HTTP/0.9` is of limited/no use.

4.5.93 SSL Client Protocols

Which SSL protocols to allow for client HTTPS/SSL connections when walking or performing results authorization, i.e. for connections from Webinator to remote `https://` URLs. The default is to leave `SSLv2` and `SSLv3` disabled, as these are known to be vulnerable to attacks. Enabling `SSLv3`, if necessary, may also require a cipher change; see note under **SSL Client Ciphers** (p. 89).

Sometimes a walker's connection fails at (or soon after) the SSL negotiation, possibly with the error message "`Missing HTTP response line in reply from` ...". This may be due to settings on the remote server that disallow certain SSL protocols – yet those protocols were enabled under **SSL Client Protocols** (e.g. for legacy reasons). In such cases, disabling various SSL protocols may enable the connection to succeed.

Note that support for some (e.g. vulnerable) protocols may end in some Webinator versions, depending on the concurrent OpenSSL libs' support: e.g. SSLv2 is no longer supported in OpenSSL 1.1.0 and later.

4.5.94 SSL Client Ciphers

Which SSL ciphers to allow for client HTTPS/SSL connections when walking, or when performing Results Authorization during searches; i.e. for connections from Webinator to remote `https://` URLs. The default (if empty) is the OpenSSL default list for the current OpenSSL client (Texis) library. Some SSL ciphers may be known to be vulnerable, and administrators may wish to disable them via this setting.

The syntax is similar to the Apache HTTP server **SSLCipherSuite** setting: an optional `SSL` (default) or `TLSv1.3` token indicating a cipher protocol group, followed (after spaces) by a colon-separated list of ciphers (OpenSSL format). Each line gives ciphers for a different protocol group, like a separate **SSLCipherSuite** Apache setting. The default (if unset/empty) is to use the OpenSSL defaults. A given cipher protocol group should not be specified more than once: combine all ciphers for a group into one line. Each distinct cipher protocol group's list is independent, and only applies to the indicated protocol(s) in the group.

Modifying – specifically, shortening – the cipher list is also a way to connect to long-handshake-intolerant HTTPS servers. These servers cannot handle an SSL `ClientHello` message longer than 255 bytes, and time out when receiving one (e.g. with `Timeout completing SSL handshake ...` errors). The default OpenSSL cipher list may cause the `ClientHello` message to exceed 255 bytes, triggering this intolerance in such servers. By setting a shorter cipher list, the `ClientHello` message can be shortened and the connection established. Disabling SNI via **SSL Use SNI** (p. 89) is another way to shorten the `ClientHello` message.

Note that support for some (e.g. vulnerable) ciphers may end in some Webinator versions, depending on the concurrent OpenSSL libs' support: e.g. 40- and 56-bit ciphers are no longer supported in OpenSSL 1.1.0 and later. Also, the list of ciphers classified as `LOW`, `EXPORT` etc. may change.

Due to increasing deprecation of weaker protocols and ciphers in OpenSSL for security, using SSLv3, TLSv1 and/or TLSv1.1 protocols when the Texis version is 8 or later may require – in addition to enabling SSLv3 via **SSL Client Protocols** (p. 88) – reducing the security level in OpenSSL. This is accomplished by adding `DEFAULT:@SECLEVEL=0` to the default (`SSL`) cipher list. Doing so is not recommended, nor is using such weaker protocols.

4.5.95 SSL Use SNI

Whether to use SNI (Server Name Indication) when walking with SSL/HTTPS. SNI enables a single-IP HTTPS server to serve the correct certificate when serving multiple hosts, and is thus required by many multi-homed name-based virtual host HTTPS servers. Disabling SNI may be useful in some circumstances to connect to long-handshake-intolerant HTTPS servers that otherwise timeout, by reducing the size of the SSL ClientHello message. Default is `Y`. Shortening the cipher list via **SSL Client Ciphers** (p. 89) is another way to work around long-handshake-intolerant servers.

4.5.96 IP Protocols

Selects which IP protocols to use for walking or other active (Webinator-initiated) fetches, e.g. Results Authorization, meta search.

4.5.97 Network Share Protocols

Specifies the minimum and maximum SMB protocols to use when accessing network shares (`file://` URLs), i.e. for walking and Results Authorization (p. 131). **SMB Min** is the minimum SMB protocol (default `SMB 2`); **SMB Max** is the maximum SMB protocol (default `SMB 3`).

Note that this setting only applies if **Network Share Access Method** is `Current` *and* in effect; it also may not appear at all if `Current` is not supported. See the **Network Share Access Method** setting (p. 90) for details.

4.5.98 Network Share Access Method

Specifies the method to use to directly access network shares (`file://` URLs), i.e. for walking and Results Authorization:

- `Current`: Use the current method, which is to use the latest helper executable, supporting SMB 1, 2, and 3. **Login Info** (p. 81) is used for credentials. The **Network Share Protocols** settings (p. 90) are in effect. No mounts are needed nor used (thus **Network Share Mounts Root**, p. 130, if available, is also ignored). HTML documents with components (e.g. frames) are not fully supported with this method: the component fetches will return errors. However the components will generally still be walked as separate URLs, if they reside on the same network share/tree as their parent.
- `Legacy`: Use the legacy method, which is to use manually-mounted shares (mounted under **Network Share Mounts Root**, p. 130, if available) for walking, and Results Authorization is only supported for Windows. Neither **Login Info** (p. 81) nor **Network Share Protocols** are used. For non-SMB/CIFS filesystems (e.g. NFS), `Legacy` must be used.

If **Proxy** (p. 82) is set, this setting has no effect, as the proxy is used instead. If Webinator is not licensed for `file://` URLs, or is on Windows (where UNC file paths are used), this setting has no effect (and will not be shown). If the platform Webinator is running on does not support the `Current` method, this setting has no effect (and will not be shown), as the `Legacy` method (or UNC file path, for Windows) is used.

When this setting is not in effect or not shown, **Network Share Protocols** (p. 90) is also not in effect or not shown.

4.5.99 Authentication Schemes

Select which authentication schemes to allow for password-protected URLs. The settable schemes are `Basic`, `NTLMv1`, `NTLMv2`, `Negotiate`, `CAS` (Central Authentication Service), `SAML/ADFS` (Security Assertion Markup Language / Active Directory Federation Services), or `File` (for `file://` URLs). NTLMv2 requires Texis version 5.01.1213917000 20080619 or later. Note that the scheme(s) actually *accepted* for a given URL are determined by the server; if none of the server-offered schemes are enabled by this setting, then the protected URL cannot be walked. This setting can be used to disable less-secure or undesired schemes, such as Basic or NTLMv1 authentication. FTP authentication is always allowed. Note that `Negotiate` authentication is only offered and supported when Webinator is running on Linux 2.6 or

later. Note also that SAML/ADFS support is limited and may not work in some environments; contact tech support in this case.

4.5.100 Embedded Security

Select the security for embedded objects on a page (e.g. frames, scripts). `Any` fetches any required object. `Non-decreasing` will fetch a required object if its security (`https://` vs. non-`https://` in the URL) is not less than the main page, i.e. an `https://` object on an `http://` page will be fetched, but not vice-versa. `Non-increasing` is the opposite. `Same protocol` requires that the protocol of the object be the same as the main page.

4.5.101 Body Storage Method

Selects the method to store the `Body` field. Choices:

- `Auto`: Automatically select best method. Typically `Blob`, but may change in future versions if methods/conditions change.
- `Table`: Store `Body` in the table. This was historically the method used before this setting was introduced.
- `Blob`: Store `Body` in a blob. This reduces the table file size, and in some situations (e.g. when **Abstract Style** is `Description`) can potentially speed up searches when large numbers of results per page are returned.

The default is `Auto`. This setting generally only needs to be changed on the advice of Thunderstone tech support.

4.5.102 Multiple Fetches

Syntax: select Y or N

`Multiple Fetches` allows a page to be fetched multiple times, and can potentially slow down a walk. It should only be used in specific situations in conjunction with `Off-Site Pages`.

For example, Consider the situation of walking two sites, `a.com` and `b.com` with `Off-Site Pages` enabled. A link from a page on `a.com` to `b.com/page.htm` is considered off-site, so it will be walked but its links won't. Then, when `b.com` starts its walk, `b.com/page.htm` won't be processed because it has already been done, causing `b.com/page.htm`'s links to not be included.

`Multiple Fetches` allows the 2nd encounter of `b.com/page.htm` to be processed again, which will allow its links to be properly processed.

4.5.103 Follow Cross-Site Links

Syntax: select Y or N

When walking multiple hosts, setting `Follow Cross-Site Links` to `Y` will allow links from one host to another to be respected, as opposed to only starting from each host's Base URLs.

If you have a lot of Base URLs that have lots of duplicate links to each other that would've been found on-site anyway, setting `Follow Cross-Site Links` to `N` can improve walk performance.

4.5.104 Max Redirects

Syntax: a whole number from 0 up or -1

This indicates the maximum number of redirects that are followed when attempting to retrieve a page. If set to -1 then redirects will not be followed when attempting to retrieve the page, but will be treated as a link.

4.5.105 Empty Form Redirects

Syntax: select Y or N

Some walked pages implement a redirect by having a HTML form that points to the target, and uses JavaScript to submit the form.

If `Empty Form Redirects` is set to `Y` and a page doesn't have any content, Webinator will treat any HTML `<form>` targets on the page as a redirect.

4.5.106 Execute Walked Dataload

Syntax: select Y or N

The Dataload system (section 5.18.5, p. 169) allows administrators to load arbitrary content into Webinator by POSTing XML files.

If `Execute Walked Dataload` is set to `Y`, then valid dataload XML files that are encountered during the walk will be fully processed as if they were uploaded to the appliance.

Dataload and replication are supported in the full Texis product, but not Webinator-only.

4.5.107 Index Name

Syntax: one or more filenames separated by space

Set the filename assumed for directory URLs. The default is "`index.html`" and "`index.htm`". This filename will be removed from stored URLs to prevent redundant fetches of the page. So the URLs "`http://www.example.com/fun/`" and "`http://www.example.com/fun/index.html`" will be considered the same and only be fetched once (as `http://www.example.com/fun/`).

Note that `Index Name` filenames are not stripped from canonical URLs within pages (as specified via `<link rel="canonical"/>`

4.5.108 DNS Mode

Syntax: choose from drop down list

This controls how Webinator looks up IP addresses for hostnames. "`Internal`" uses Texis's own internal parallelizing name lookup routines. "`System`" uses the standard system routines. You should use "`Internal`" unless it causes compatibility problems.

4.5.109 Net Mode

Syntax: choose from drop down list

This controls what API Webinator uses to access Web pages. "`Internal`" uses Texis's own internal parallelizing Web fetch routines. "`System`" uses the standard system routines. You should use "`Internal`" unless it causes compatibility problems.

Note: "`System`" only has effect for the Windows version of Webinator. It does not currently support parallel access and some other Web features of the "`Internal`" mode. However, it does provide an alternate way to access NTLM-controlled sites (using the user/password set in `Login Info`), in versions prior to October 2004. Later versions support NTLM authentication in the default "`Internal`" net mode.

4.5.110 User Agent

Syntax: full user-agent string

Set the `User-Agent` (browser type) to report to web servers. Normally Webinator reports itself as "`Mozilla/4.0 (compatible; T-H-U-N-D-E-R-S-T-O-N-E)`". Modify this setting to report as a different user agent. If you want to emulate a particular browser, you can access your site with that browser, then check the site's transfer log to see what user agent string was logged (typically the last double-quoted entry on the line).

4.5.111 Robots.txt Agents

Syntax: one or more user-agent strings, one per line

This is a list of user agents to respect when checking `robots.txt` on a site. The `robots.txt` group with the `User-agent` string that is a case-insensitive substring of the earliest agent listed in **Robots.txt Agents** will be used; i.e. the **Robots.txt Agents** should be listed highest-priority first. If multiple `robots.txt` groups match the same agent, the group with the longest substring-matching `User-agent` is used. If no agents match, and a group for agent "`*`" is present, it is used. The default value for this setting is "`webinator`".

For example, changing this setting to `MyBot` and `Googlebot` and given this `robots.txt` file:

```
User-agent: Google
Disallow: /some/google/dir

User-agent: MyBot
Disallow: /some/other/dir
```

then Webinator will not walk `/some/other/dir`, but will still walk `/some/google/dir`: while both agents substring-match, and `Google` is a longer substring, `MyBot` is listed first in **Robots.txt Agents** and is thus higher priority.

Given this `robots.txt` with the same setting:

```
User-agent: Google
Disallow: /some/google/dir

User-agent: Googlebot
Disallow: /some/bot/dir
```

then Webinator would not walk `/some/bot/dir`, because while both agents substring-match `Googlebot`, `Googlebot` is the longer match.

4.5.112 Mime Types

Syntax: one or more acceptable MIME types, each on a separate line

These are the Multipurpose Internet Mail Extensions (MIME) types that Webinator informs the web server are acceptable. MIME types have the syntax `type/subtype`. Either `type` or `subtype` may be `*` to mean "any". By default all MIME types are allowed (`*/*`).

4.5.113 Custom Headers

Syntax: one or more Name/Value pairs

These are extra headers to send to the webserver when fetching pages. Sometimes a server application will not work right when optional HTTP headers are not present. This allows setting them as needed in such situations.

Header names should not have any spaces or the trailing colon (:).

4.5.114 Respect Expires Header

Syntax: choose from drop down list

For `Refresh`-type walks, this controls how the `Expires` header is used. Set to `No` the `Expires` header will be ignored. Set to `Limited` the `Expires` header will be used, but limited by the **Minimum Refresh Time** and **Maximum Refresh Time**. Set to `Yes` the `Expires` header will be treated as definitive.

Invalid and out of range headers will be ignored, with the exception of "0".

4.5.115 Cache Content

Syntax: choose from drop down list

Cache Content allows Webinator to store a copy of the content as it walks. In the search interface, a `View Cached` link will appear by results, which allows users to download the contents directly from Webinator rather than using to the original location.

Matches in the content will be highlighted in cached pages that are HTML documents. The highlighting uses the **Context Highlighting** setting on the search settings page to style the highlights.

In addition caching content can be useful in situations where the original location is unavailable, either because a server is down or the user does not have access to the file server from a remote location.

The choices for **Cache Content** are:

- `None` *(default)* - No extra storage of the original content occurs, Webinator only maintains the plain text representation of the results for searching (as it always has).
- `Results Only` - While walking, Webinator maintains a copy of the original content. Selecting `View Cached` from a search result will present that cached copy. Links and references from web pages are left unmodified, pointing back to the original site.
- `Site Mirror` - While walking, Webinator maintains a copy of the original content, and also collects and stores auxiliary content used by web pages (images, CSS stylesheets, etc) that isn't normally necessary for Webinator walks.

  When search users select `View Cached`, links and references on cached web pages are rewritten to point back to Webinator, utilizing the cached versions of these resources. This allows a full cached view of a site to operate when the original site becomes unavailable, instead of presenting a cached web page with lots of broken images and links.

  Note that some complex, dynamic web sites might not be able to be fully mirrored.

4.5.116 Default Refresh Time

Syntax: choose from drop down list

For `Refresh`-type walks, this is the default time period to initially try refreshing a URL; typically set to 1 minute. Note that the actual refresh period is dynamically computed for each URL based on how often it changes.

4.5.117 Minimum Refresh Time

Syntax: choose from drop down list

For `Refresh`-type walks, this is the minimum time period to try refreshing a URL. The actual refresh period is dynamically computed for each URL based on how often it changes, and will not be less than this value. This prevents too much time being spent refreshing a very dynamic page (i.e. constantly refreshing it and loading the web server). Typically set to 1 minute.

4.5.118 Maximum Refresh Time

Syntax: choose from drop down list

For `Refresh`-type walks, this is the maximum time period to try refreshing a URL. The actual refresh period is dynamically computed for each URL based on how often it changes, and will not be greater than this value. This ensures that all URLs – even relatively static ones – are eventually checked for changes.

4.5.119 Maximum Process Size

Syntax: choose from drop down list

Upper limit to memory size of walker processes. If a walker process exceeds this limit, it is re-started (at the same point it left off) by the dispatcher, at most once. If the same child repeatedly exceeds this limit, the walk may stop until it is re-started via schedule or manually. Note that this is a soft, not hard limit: the process may exceed it briefly, yet will exit gracefully and not crash if so.

4.5.120 Replication Settings

Syntax: List of hosts and profiles

A list of hosts and profiles to send walk data updates to. The hosts must have the sending server listed as one of the **Cluster Members** under **System Wide Settings**.

Replication is supported in the full Texis product, but not Webinator-only.

4.5.121 Send Data

Syntax: select Y or N

Send Data will cause this profile to send changes in searchable content to be sent to this replication target. This includes adding or changing content, and refresh walks removing URLs that are no longer present.

This does not include starting a new database for a "New" walk. If URLs are present in a previous walk and not present in the next "New" walk, this will not remove the URLs from the replication target. This requires `Send Settings`, as described below.

4.5.122 Send Settings

Syntax: select Y or N

Send Settings causes this profile to send any settings changes to this replication target. This includes starting "New" walks, which will cause the replication target to start from a fresh database like the replication sender does.

4.5.123 Batch Rows

Syntax: number

Defines the number of items the replication sender will attempt to accumulate in a single batch. This should not need changed.

4.5.124 Batch Size

Syntax: number (in bytes)

Defines the threshold for size for a sending a batch of replication items. When collecting items to replicate, once the size of the items is over this threshold, the batch will be sent off to the targets. This should not need changed.

4.5.125 Batch Idle

Syntax: number (in seconds)

Defines the idle timeout for sending a batch of replication items. When collecting a batch, if no new items appear for this many seconds, the replication batch will be sent to the targets. This should not need changed.

4.5.126 Log Replication

Writes information for this profile's replication queue processor to `replication.log`.

If both the System-wide `Log All Replication` and this profile's `Log Replication` are set, logging for this profile will be the more verbose of the two.

See also "Replication" 5.18.

4.6 Search Settings

This group of options applies to the standard search and provides a convenient way to make common changes to the search behavior and appearance. You are not limited to the features listed here. You may modify the search script to look however you want and to behave however you want.

See also "Customizing Webinator's Appearance" 3.5.

4.6.1 Notes

This is the same setting as **Notes** under Walk Settings: a scratch pad area for the administrator of the profile. It in no way affects the walk or search.

4.6.2 Query Logging

Syntax: select Yes or No button

This indicates whether the search should log user queries. If Yes, users' queries are logged to the querylog table of the database. The contents of this table may be viewed from the `Query Log` menu of the Administrative Interface.

Note: The query log table gets erased during every new walk. You will only be able to view queries that have occurred since the latest new walk. Refresh walks do not cause the table to be erased.

4.6.3 Rotate Schedule

Syntax: The day of week (or daily) and the time of day to rotate

This selects when to rotate query logs on this profile. During a rotate action, the log table data is optionally emailed to someone, and then the data is erased from the log table.

See also `Attach Logs` (section 4.5.3).

4.6.4 Email

Syntax: A valid email address

When the query log is rotated (according to the schedule set), an email message with an attached file (containing the previous log data) is sent to this address. Multiple addresses may be specified, separated by commas.

4.6.5 Result Order

Syntax: select Relevance, Date, or URL button

This determines the default ordering of search results.

- `Rank` - search results are ordered by rank (or relevance) by default.
- `Date` - search results are ordered by date descending (newest first) by default.
- `URL` - search results ordered by their URLs alphabetically by default.

Search users may select the alternate ordering from this default in the Advanced search form. Note that more ordering options are possible by setting the `order` query variable directly; see discussion under **Search Parameters** - **Search control** (p. 151). Note also that the factors that influence a document's rank for `Rank` ordering – including date – can be controlled; see **Word Ordering** and other rank "knobs", p. 123.

4.6.6 Results Style

Syntax: choose from drop down list

This controls the style used for displaying individual results to user queries. There are various styles from which to choose. The arrangement and amount of information varies in every style. In the administrative interface you may click the question mark (`?`) next to **Results Style** to see a sample of each of the available styles.

4.6.7 Allow RSS

Syntax: select Yes or No button

If `Allow RSS` is set to `Y`, then each search result page will include a reference to an RSS feed for that search, which users will be able to monitor.

Setting `Allow RSS` to `N` will both remove the reference from search result pages, and disallow the viewing of RSS feeds.

4.6.8 Format XSL Output

Syntax: select Yes or No button

If set to `Y`, then extra line breaks are added in to the output of the server-side XSL stylesheet processing. This has the following effects:

- It makes the HTML output more readable by humans, changing it from one extremely long line to a well formatted document.
- It adds a small amount of size to the document (usually between 1-4%)
- Adding line breaks at certain locations can sometimes trigger odd rendering bugs in Internet Explorer (adding spaces where there shouldn't be spaces).

4.6.9 XSL File

Syntax: Browse local disk for a XSL file

This allows the use of a customized XSL file to format the output of a search. A default XSL style sheet is included with Webinator (`/webinator/xsl/default.xsl`). The **XSL File** option is used only if the

Results Style is set to `XSL Stylesheet`. The links below this option display the current XSL stylesheets, which may be downloaded for editing and then re-uploaded with this option.

Note that the `/webinator/xsl` subdirectory of the web server's document root must be writable by the Texis user in order for this option to work. The install program normally does this at installation however.

4.6.10 Abstract Style

Syntax: choose from drop down list

This setting controls the short description or abstract that is generated for each search result. Choosing `Query` uses a snippet that matches the query. `Beginning` uses the start of the document's content. `Top` uses the top of the current page. `Description` uses the value of the `Description` meta tag.

4.6.11 Abstract Length

Syntax: enter number in text box

This determines the length in bytes of the document abstract.

4.6.12 Max Title Length

Syntax: enter number in text box

This determines the maximum length in bytes of the document title shown in the results. If the title is over this length, it will be truncated and ended with ellipses.

Title length may be expanded up to 10 characters over this setting in order to avoid cutting off in the middle of a word.

Set to `-1` to always use the full title.

4.6.13 Max URL Display Length

Syntax: enter number in text box

This determines the maximum length in bytes of the matching URL shown in the results. If the title is over this length, it will be truncated after the hostname with ellipses and ended with as much of the path and filename as it can.

Note that this does not affect the URL that is actually linked to - that URL is always the full, proper URL. This setting only affects the displayed URL.

Set to `-1` to always use the full URL.

4.6.14 Results per Page

Syntax: a whole number

This controls the default number of results (hits) listed on each results page. When there are more than this many results to a user's query the user will have to hit "next" to see more results.

This number may be overridden by the search user with the `rpp` URL/query parameter – up to a maximum of **Max User Results per Page** – to allow the search user to customize the number of results per page.

Note that increasing the results per page can increase the time needed per search, as more information must be retrieved due to more results.

4.6.15 Max User Results per Page

Syntax: a whole number, or -1 to disable

Search users are able to customize how many results per page they see – which defaults to **Results per Page** – by supplying the URL/query parameter `rpp`. The **Max User Results per Page** setting places an upper bound on how large `rpp` may be set to, as increasing the results per page can increase the time (and load) per search, since more information must be retrieved for more results.

If **Max User Results per Page** is set to `-1`, then the `rpp` parameter is ignored, i.e. the user may not override **Results per Page**.

4.6.16 Page Links Shown

Syntax: a whole number, defaults to `10`

This specifies the number of page links to include in the summary of the results.

For example, if we are on page 22 of 5,000 total results, by default direct links will be shown to pages 18 through 27 (for a total of 10 links). If `Page Links Shown` is set to `20`, it will show links 13 through 32, for a total of 20 page links.

4.6.17 Results per Site

Syntax: an integer select box and Yes/No button

The **Max** setting controls the maximum results per site per page to display. For large profiles with many pages (and thus results) per site, setting **Results per Site** can increase the results variety shown on a single page, by replacing some same-site results with lesser-ranked but different-site results from subsequent pages.

When results are limited to N per site, no more than N results for any given site will be shown on any given page. Results past N for a site are suppressed, and results from new sites (that are not past N yet) added, until the page is full. Once the page is complete, the results are reordered: the second and later results from a site are moved up under the first result from that site, indented, and followed by a `More results for`

`site` link. The next page's results will be obtained by the same process, resuming at the point in raw results left off by the previous page. Note that a given site may appear on subsequent pages (if there are more results for it), but its results that were suppressed for the previous page will not be shown (because they are visible via the previous page's `More results for site` link). Also, since there are now two degrees of navigation through the results – standard pagination, plus the `More results for site` links – not all of the total results counted may be shown via pagination: the rest are shown via `More results for site`.

Note also that setting **Max** to other than `Unlimited` can increase search time, as potentially much more than one page of raw results must be obtained and suppressed, and results must be regrouped.

The **Allow override** button controls whether the search user can override the profile's **Max** limit on the Advanced Search form (via the `sr` query string variable). This can be set to `N` to prevent the potential delay of grouping by site, or `Y` to allow the user to set a custom value. For profiles that are Meta Search back-ends, if the front-end Meta Search is using Results per Site, all the back-ends should have **Allow override** set to `Y` so that the front-end's value can be used, for consistency.

4.6.18 Allow site: syntax

Syntax: a Yes/No button

This controls whether to allow the `site:`*host.domain* query syntax in a search, to limit results to a single domain.

For example, to search for the words `panda bear` but only on sites in the `example.com` domain, enable the syntax and use the query:

```
panda bear site:example.com
```

This will return results from `example.com` as well as e.g. `www.example.com`, but not `example.com.us`.

No space may appear before or after the colon, nor in the domain. A `site:` clause may only be used in conjunction with other results-producing query parameters, e.g. keywords.

A `site:` clause will override any value in the **From this domain** box on the Advanced Search form, which uses a separate variable (`sq`). For profiles that are Meta Search back-ends, if the front-end Meta Search is using the `site:` syntax, all the back-ends should have **Allow site: syntax** set to `Y` so that the front-end's value can be passed in.

Note that a `site:` query requires post-processing, which may reduce query performance. For this reason, **Allow Post-Processing** (p. 117) must be enabled as well for such queries.

4.6.19 Allow link: syntax

Syntax: a Yes/No button

This controls whether to allow the `link:`*URL* query syntax in a search, to find results that link to the given URL. No space may appear before or after the colon, nor in the URL (unless URL-encoded).

For example, the query:

```
link:http://www.example.com/dir/page.html
```

will list all results that link to `http://www.example.com/dir/page.html`.

Combining a `link:` clause with any other clause in the query (e.g. keywords) may reduce search performance, due to possible post-processing. For this reason, **Allow Post-Processing** (p. 117) might need to be enabled as well for such queries.

4.6.20 Results Width

Syntax: a whole number or a percentage valid for an HTML <TABLE> WIDTH

This controls the width of the `<TABLE>`s used in the search results. This may be a number indicating a fixed width or a number from 1 to 100 followed by a percent sign(`%`). This tells the user's web browser how wide to make the table.

4.6.21 Box Color

Syntax: a color name or number valid for HTML color specification

This controls the color of the "gray" informational boxes at the top and bottom of search results pages.

4.6.22 Show File Icons

Syntax: select Yes or No button

Show file type icons on individual results. These icons appear for non-html types (office files, PDF, images, etc). The exact positioning or usage of the icons can be customized using the XSL Stylesheet (see `XSL File` setting 4.6.9).

4.6.23 Show Advanced Search

Syntax: select Yes or No button

This controls whether or not the Advanced Search button is displayed on the search form. If set to No then the button will be hidden, otherwise it will be displayed.

4.6.24 Query Autocomplete

Syntax: select Yes or No button

Query Autocomplete (also known as "typeahead") lets the Webinator suggest full words from the user's partially typed queries as they're being typed. If the user has typed `ense`, query autocomplete may suggest `ensemble`, `enseignement`, etc.

Like Spell Check, the list of words used for Query Autocomplete comes from the current profile's content, so the only suggestions made will be words that occur somewhere in the content.

Query Autocomplete is ordered with a priority system, where words that occur more often are ranked higher, and words that occur in titles are ranked much higher.

4.6.25 Max Completions

Syntax: a whole number

This controls the maximum number of completions that will be shown to the user when using Query Autocomplete functionality.

4.6.26 Results Highlighting

Syntax: select None, Classes, Inline or Bold

The user's query will be highlighted in various parts of the search results (Title, Abstract, etc.) with the selected method:

- `None` or `N` - No highlighting will be done in search results.
- `Classes` or `Y` - Terms will be highlighted with `<span>` tags that refer to classes that are defined in a separate CSS file, `/webinator/common/search.css` by default. Each term in the query is tagged with a different class, which are each highlighted a different color in the default `search.css`. All these rules can be overridden with your own CSS on the results page.
- `Inline` - Terms will be highlighted with `<span>` tags that directly specify a fixed CSS style. This is not customizable, but is self-contained and does not depend on a separate stylesheet or file. Same visual result as `Classes` with the default CSS.
- `Bold` - Terms will be highlighted with `<b>` tags.

The default is `Bold`.

4.6.27 Context Highlighting

Syntax: select None, Classes, Inline or Bold

The user's query will be highlighted in the context view (Match Info page) as well as the cached content view with the selected method. Same choices as for **Results Highlighting**.

The default is `Classes`.

4.6.28 PDF Query Highlighting

Syntax: select Yes or No button

When making links to PDFs in search results, Webinator will add extra info to the link which will cause the user's query to be highlighted by the PDF viewer. Changing this setting to "N" will remove that extra information from the link, and no longer highlight the user's query in the PDF document.

4.6.29 PDF Highlighting Format

Syntax: select "Acrobat 7+" or "Legacy"

Controls the format used to provide query highlighting information for PDF links.

- `Acrobat 7+`: Uses the `search` parameter as defined in the PDF Open Parameter specification. Supported by Adobe Acrobat 7 (January 2005) and all later versions.
- `Legacy`: Uses the `xml` Highlight File Format syntax, which has been deprecated by Adobe. It was disabled by default in Acrobat Reader 9 (July 2008), and completely removed from Acrobat Reader X (Nov 2010) and all later versions.

This is a temporary setting is for compatibility, and will be removed in a future version of the Webinator (forcing `Acrobat 7+` behavior).

4.6.30 Font

Syntax: a font name valid for HTML <FONT> specification

This specifies the font to use throughout the search interface.

4.6.31 Display Charset

Syntax: a standard IANA charset name

This sets the charset used to display search results in. The default if empty is the charset for Storage Charset under All Walk Settings. This charset should be a superset of `US-ASCII` (same 7-bit sequences), compatible with Top HTML, and translatable by Webinator from Storage Charset.

A `<META HTTP-EQUIV=Content-Type>` tag in Top HTML will be updated automatically to reflect this charset. This update can be disabled by putting 2 or more spaces between `META` and `HTTP-EQUIV` in Top HTML.

Note that if the Display Charset differs from the Storage Charset, search results must be converted on-the-fly, potentially degrading performance slightly. Thus, if Display Charset is ever changed, it is recommended that Storage Charset be changed as well, and after the next rewalk (when all the database data is now in the new Storage Charset), Display Charset be change back to default (empty, which will still display in the new Storage Charset).

4.6.32 Top HTML and Bottom HTML

Syntax: HTML

This is static HTML to place at the beginning and ending of every search page respectively. It is useful for setting styles and displaying navigation menus and otherwise making the search pages look like the rest of your site.

The `<!-- THUNDERSTONE_HEADERS -->` placeholder is replaced at search time with custom information necessary for search. If you customize your Top HTML, make sure `<!-- THUNDERSTONE_HEADERS -->` is somewhere in the Top HTML's `<head>` section.

Top and Bottom HTML when placed together should be exactly what is required to create a complete and valid HTML page. You can use your favorite HTML editor to create a page with a placeholder for the search form and results. Then cut and paste the section of HTML before the placeholder into the Top HTML and the section of HTML after the placeholder into the Bottom HTML.

If `$query` occurs within these fields, it will be replaced by the user's query.

4.6.33 Enable Sherlock

Syntax: select Yes or No button

This informs the search to include comment tags in the results page to allow Sherlock to process the list.

Sherlock is a metasearch tool for Macintosh computers.

4.6.34 Best Bet Match Mode

Syntax: select from two options

Controls how best bets keywords are matched with the user's query.

With `Show when search query is contained in Best Bet keywords` mode (the default), a best bet is matched if the user's query is contained in the Best Bet's keyword(s). Webinator internally does a "search" with the best bet keyword(s) as content, and the user's query is the search.

With `Show when Best Bet keywords are contained in search query` mode, a best bet is matched if the Best Bet's keyword(s) are contained in the user's query. Webinator internally does a "search" with the user's query as content, and the best bet keyword(s) are the search terms.

Examples:

With `Show when search query is contained in Best Bet keywords`, a best bet with the keywords `pay raise` will be triggered for the search queries `pay`, `pay raise`, or `raise`, because the user's query is contained in the best bet keywords. But it will *not* be triggered if a user searches for `pay schedule` or `pay raise schedule`, because the user's query is *not* fully contained in the keywords.

With `Show when Best Bet keywords are contained in search query`, a best bet with the keywords `pay raise` will be triggered for the search queries `pay raise` or `pay raise`

`schedule`, because the best bet keywords are contained in the user's query. But it will *not* be triggered by `pay`, `raise`, or `pay schedule`, because the keywords are *not* fully contained in the user's query.

4.6.35 Top Best Bet Title

Syntax: text

This is the title text of best bets displayed above the search results. Common choices are "`Best Bets`" and "`Suggested Links`". See `Using Best Bets` 5.16 for more details.

4.6.36 Right Best Bet Title

Syntax: text

The title text of best bets displayed to the right of search results. Common choices are "`Best Bets`" and "`Suggested Links`". See `Using Best Bets` 5.16 for more details.

4.6.37 Top Best Bet Group

Syntax: choose group from drop-down list

This controls which group of best bets will be shown above the results. The group must already be created. See `Using Best Bets` 5.16 for more details.

4.6.38 Right Best Bet Group

Syntax: choose group from drop-down list

This controls which group of best bets will be shown to the right of the results. The group must already be created. See `Using Best Bets` 5.16 for more details.

4.6.39 Top Best Bet Box Color

Syntax: valid HTML color

This controls the color to be used for the background of the top best bet box. See `Using Best Bets` 5.16 for more details.

4.6.40 Right Best Bet Box Color

Syntax: valid HTML color

This controls the color to be used for the background of the right-side best bet box. See `Using Best Bets` 5.16 for more details.

4.6.41 Top Best Bet Border Style

Syntax: select from drop-down list

This controls the style of the top best bet box border. You can choose to have no border, a border around all the best bets, or an individual border around each result. See `Using Best Bets` 5.16 for more details.

4.6.42 Right Best Bet Border Style

Syntax: select from drop-down list

This controls the style of the right-side best bet border. You can choose to have no border, a border around all the best bets, or an individual border around each result. See `Using Best Bets` 5.16 for more details.

4.6.43 Right Best Bet Box Width

Syntax: enter number in text box

This controls the width of the best bet boxes shown to the right of the regular search results. See `Using Best Bets` 5.16 for more details.

4.6.44 Authorization Method

The `Authorization Method` setting controls what Results Authorization method(s) are used by Webinator when verifying user access to search result URLs. See the Results Authorization section (p. 131) for details. The possible settings are:

- `None`: No access verification; return all search results to all users. This is the default. It is also the setting that should be used for a Meta Search profile, even if one or more of its back-end profiles does use **Results Authorization**: the request and response for credentials will automatically be passed back and forth from front-end Meta Search to back-end profiles, which will handle the authorization (not the front-end).

- `Forward login cookies`: Webinator will forward login cookies from the user to the result URL. This is for custom HTML-form-based single-sign-on systems.

- `Basic/NTLM/file - prompt via form`: Webinator will prompt the user for their credentials with a form, then send them to the result URL via HTTP Basic, NTLM or Windows/SMB file authentication.

- `CAS`: Webinator will use the Central Authentication Service to proxy user credentials. The **Login URL** must be set to the CAS login service, with a `service` parameter pointing back to Webinator. Additional caveats apply; see details under **Login URL** for CAS, p. 109.

4.6.45 Login Cookies

For the `Forward login cookies` Results Authorization method, one or more cookies must be named in the `Login Cookies` setting. No values are given, as they will be obtained automatically on a per-search basis from the user.

When a user conducts a search, if the named cookies are seen from the user's browser, the user is assumed to be logged in, and the cookies are forwarded to the result URLs for authorization. If the named cookies are not seen, the user is assumed not to have logged in yet, and is redirected to `Login URL` instead.

Note that these cookies *must* be set by their server with `Domain` and/or `Path` attributes that let them be sent to the Webinator's domain name and `.../search` path. Otherwise the user's browser will not send them to Webinator (for forwarding), and thus Results Authorization will not work.

4.6.46 Login URL

For the `Forward login cookies` **Authorization Method**, when none of the **Login Cookies** are seen at search time(or for the `CAS` method when no service ticket is seen), the user is assumed not to have logged in yet. In this event, the user will be redirected to **Login URL**, which should be the URL to the site's form-based (or CAS) login page.

After logging the user in, the site's login page should redirect the user back to their original search. To accomplish this, the special token "`%REFERER%`", if used in the **Login URL**, will be replaced with the URL back to the user's search. Thus, it could be assigned to a query-string variable in the **Login URL** so that the login page can redirect back to the search. E.g. with this value for the **Login URL**:

```
http://login.example.com/login.asp?searchurl=%REFERER%
```

Webinator would redirect the user to `http://login.example.com/login.asp`, with the `searchurl` variable set to the Webinator search page (with query). The `/login.asp` code should be configured to then redirect the user back to the `searchurl` query variable after login.

Additional CAS Setup

For the `CAS` **Authorization Method**, the **Login URL** must usually be HTTPS (a CAS server requirement). It also must point to the actual CAS login service, not a wrapper. This is because Webinator will also map the `/login` part of the URL to `/serviceValidate` and other standard CAS services for ticket validation after login. Thus a URL such as `https://cas.example.com/cas/login?service=%REFERER%` should be used for **Login URL** for CAS.

The CAS server must also be configured to work with Webinator. When configuring, be sure to use a URL pattern that matches all possible Webinator search and admin URLs, e.g. one that matches at least `https://webinator.example.com/texis/webinator/`.... Consult your CAS server documentation for how to configure these items:

- Webinator must be allowed to use CAS. This typically involves ensuring its URLs (see above) match a list or pattern of permitted URLs. For an Apereo CAS server, this may involve ensuring the `serviceId` setting of the appropriate config file (e.g. `HTTPSandIMAPS-10000001.json`) matches Webinator URLs. Lack of permission may result in an error such as "`Application Not Authorized to Use CAS`" from the CAS server when the user attempts to search, and is redirected to the CAS login.

- Webinator must be allowed to proxy. For Apereo CAS, this may involve setting a `proxyPolicy pattern` (e.g. via JSON). Lack of proxy permission may result in an error such as `INVALID_PROXY_CALLBACK` from Webinator during searches.

- All CAS-protected services that may be walked and appear in Results Authorization search results must allow Webinator to proxy them. For Apereo CAS, this may involve setting the `allowedProxyChains` parameter in the CAS Validation Filter. Lack of this permission may result in these services always being rejected (via HTTP 500 Server Error) as unauthorized, and not shown in search results.

- Depending on the CAS server's configuration, Webinator may have to be accessed via an HTTPS/SSL URL.

- The CAS server may also need to trust Webinator's SSL certificate, i.e. have that certificate's CA in its trust store. Lack of trust may also result in an `INVALID_PROXY_CALLBACK` error.

If encountering problems configuring CAS with Results Authorization, be sure to check the CAS server log files for information that may help diagnose the issue. Also note that Results Authorization with CAS is not currently supported for Meta Search.

4.6.47 Basic/NTLM/file Cookie Type

For the `Basic/NTLM/file - prompt via form` Results Authorization method, this setting controls what cookie type to use for Webinator's copy of the user's credentials.

With `Basic/NTLM/file - prompt via form` set, when a user conducts a search for the first time, a form is presented (from Webinator) asking for a user and password. The user/pass is sent back to the user as a cookie from Webinator for use in future searches without having to re-prompt. The user/pass is also simultaneously used to validate search results via HTTP Basic/NTLM or Windows/SMB file access.

The `Basic/NTLM/file Cookie Type` setting controls whether this cookie from Webinator should use the `Login Expiration` setting from System-Wide Settings (which itself can be set to `Session` or a custom duration), or `Session` (discarded after browser closure for security).

Note that the cookie for `Basic/NTLM/file Cookie Type` is distinct from the `Login Cookies`; they are used for different access methods. The former originates from Webinator and is only ever sent to/from the user and Webinator: non-cookie-based access methods are then used from Webinator to the result URLs for actual authentication. `Login Cookies`, however, originate from a third-party form-based login system, and pass from the login server to the user to Webinator to the result URLs.

4.6.48 Login Verification URL

When doing Results Authorization, Webinator does not validate credentials or cookies on its own. They are passed along to the content server, who decides whether the individual results are allowed or denied.

Since authentication is handled by another server, when search results are denied access, Webinator cannot know if the denial is URL-based (lack of access by the user), or login-based (mistyped/wrong password).

To differentiate the two and give users a chance to correct mistyped passwords, a `Login Verification URL` may be set. This should be a URL that *all* users have access to, but that is still protected (i.e. anonymous users are denied). It should be an actual file (not a directory), preferably small (a few KB), and permanent (not likely to move, be renamed or have perms changed).

If `Login Verification URL` is set, Webinator will verify a user's prompted-for login by accessing this page. Since all users have access to it, a denial is assumed to mean the login was incorrect, and the user will be re-prompted for their credentials. Without a `Login Verification URL` set, a mistyped password will result in no search results, but the user will not know if they do not have access to the results, or they merely mistyped their password.

`Login Verification URL` can also be useful with the `Forward Login Cookies` Results Authorization method, when used in conjunction with an `Authorization Target` of `Login Verification URL Only`, as described below.

4.6.49 Authorization Target

When using Results Authorization, individual results will be checked to ensure the search user has access to them. This can be wasteful if you know your entire results use the same permissions, e.g. if a user can access one thing, they can access everything.

You can set `Authorization Target` to `Login Verification URL Only`, and if the `Login Verification URL` check is successful, Webinator will assume all the individual results are allowed and skip authorizing them individually.

4.6.50 Unauthorized Result Query

For all `Authorization Method` types of Results Authorization, it is assumed a protocol-level denial will be issued when Webinator accesses URL(s) that a user does not have access too. E.g. for HTTP URLs, a `401 Unauthorized` message should be issued.

However, some servers may only issue a human-readable denial message, but otherwise return an ok (e.g. HTTP `200`) protocol message. For such results Webinator will assume the user has access, and will erroneously return the result.

To remedy this, `Unauthorized Result Query` may be set to a query that will match only denied pages (e.g. "`Access Denied`"). The `Field/Type` box should be set to the query type (substring vs. REX) and field (raw HTML vs. formatted text) for the search. The `Query` field is set to the actual substring or REX query. See p. 193 for details on REX search syntax.

Note that this setting imposes an extra search load, as each search result must be verified with a full-page GET instead of a HEAD, as well as queried against. Thus, `Unauthorized Result Query` should only be set if absolutely necessary.

4.6.51 Username Fixup

Username Fixup allows you to make modifications to the Results Authorization username provided, such as adding or removing a domain. This allows multiple back-ends with slightly different authentication schemes to be searched simultaneously in a Meta Search.

- `Search` - the search expression to match on the incoming username. Unless you're stripping off a domain, this should be left blank to match everything.
- `Replace` - the replacement string used to modify what was matched in the search. Please see examples below, or the **Replacement Strings** of the Vortex manual on our website for the exact syntax.

For example, suppose you have a wiki and a file server. They use the same authentication back-ends, but the wiki takes the format `username` and the file server takes the format `DOMAIN\username`. If you create a profile for each of them and set the `Username Fixup Replace` value for the file server to `DOMAIN\\\1`, then you can meta-search both with `username` and each will get the format it needs.

Examples

- Changing `username` to `MYDOMAIN\username`
  - Search - *(Empty)*
  - Replace - `MYDOMAIN\\\1`
- Changing `MYDOMAIN\username` to `username`
  - Search - `>>=!\\+\\=.+`
  - Replace - `\4`
- Changing `MYDOMAIN\username` to `OTHERDOMAIN\username`
  - Search - `>>=!\\+`
  - Replace - `OTHERDOMAIN`

4.6.52 Max Docs to Auth-Check

This setting is the maximum number of raw (pre-auth-check) search result URLs to examine for authorized results, during results authorization. Decreasing this limit can speed up searches and reduce origin server load, at the cost of possibly truncated displayed results. E.g. noisy queries that match many overall

documents on the server, but few of which are authorized for the search user, may use a lot of server resources, so reducing this limit may reduce that load.

The maximum value is -1 or blank (the default), for no limit: i.e. continue until all results are checked, or `Successful Auth Result Limit` or `Total Auth Timeout` is reached.

4.6.53 Successful Auth Result Limit

This setting is the maximum number of authorized (displayable, post-auth-check) results to try to establish, during results authorization. Increasing this limit makes it more likely to get an exact hit count for a search (instead of a single page), at the expense of more search time and more origin server load.

The minimum (and default if empty) is the same as the `Results per Page` setting (p. 101), which produces a page of results the fastest. The maximum is -1 for no limit, i.e. continue until all results are checked, or `Max Docs to Auth-Check` or `Total Auth Timeout` is reached.

4.6.54 Total Auth Timeout

This setting is the maximum total time in seconds to spend searching and authorizing results, during Results Authorization. The maximum setting value is -1 for no limit, i.e. let `Search Timeout` (p. 120) cancel the search if reached. Any other negative value is relative to `Search Timeout`. Thus the default (if empty) of -5 means stop searching 5 seconds before `Search Timeout`, so that there are a few seconds left to send the results to the user.

4.6.55 Allow Authorization URL

If enabled, the `Authorization URL` field of each document is used for Results Authorization instead of the document URL. (If the `Authorization URL` field of a document is empty, or this setting is disabled, the document URL is used.) Enabling this can speed up searches under certain circumstances.

Sometimes an entire group of documents share the same authorization. For example, on some systems the contents of a directory always have the same authorization as the directory itself. In other words, every user's permissions on the files in any directory is the same as their permissions on the directory itself. If this is the case, then Results Authorization can authorize all results in the directory just by authorizing the directory itself, once. This reduction in calls speeds up searches.

For this optimization to be effective, the `Authorization URL` field in the database must be populated (see **Data from Field**, p. 62). For example, on systems where the contents of a directory always have the same authorization as the directory itself, `Authorization URL` should be set to the parent dir of each URL. The more files there are (on average) in a given directory, the more effective this optimization will be. Additionally, the **Authorization Caching** setting should be set to `Session`, so that the one-time directory authorization can be reused for each result inside the directory. (Otherwise Results Authorization must repeat the directory authorization for every result in the directory, as normal.)

The `Authorization URL` field may also be used on systems that do not meet the group-authorization criteria (many docs sharing the same authorization) detailed above. An environment may exist where the

walked/result URL is simply not the same URL that should be used for Results Authorization. For example, the walk/result URLs may be `file://` URLs, yet the authorization should take place with `http://` URLs of the same host and path. In such a case, the `Authorization URL` field could be populated with the `http://` variant to tell Results Authorization to use those URLs. In this instance, the field is being used to properly authorize URLs, and will not necessarily speed up searches (because the `Authorization URL`s are unique and not shared across groups).

4.6.56 Authorization Caching

Whether and how to cache Results Authorization traffic. The default of `None` does no caching. When set to `Session`, Results Authorization traffic is cached for the duration of the session, i.e. that search alone. Normally caching is of little benefit, because authorization URLs are typically the same as result URLs, and the latter are typically unique in a given search; thus caching will not help. However, if the `Authorization URL` field is populated, and **Allow Authorization URL** is enabled, enabling caching may speed up Results Authorization searches. See **Allow Authorization URL** (p. 113) for details.

4.6.57 Debug Results Authorization

Enabling this setting causes copious debugging information to be logged. It should only be enabled at the request of Tech Support for diagnosing Results Authorization problems.

4.6.58 Show Authorization Info

Enabling this causes details about the Results Authorization process to be displayed on the search results page - which URL are being attempted, what the outcome is, how long it takes, etc. This can assist in troubleshooting why results aren't displaying when expected.

- `None` *(default)* - No information is displayed.
- `Admin Users Only` - information is displayed only if the browser is currently logged in to the admin interface. This allows admins to troubleshoot Results Authorization without exposing information to all users.
- `All Users` - information is displayed for all search users.

WARNING - The information shown includes info about URLs that search users don't have access to (explaining how/why they failed). The Webinator acknowledging the existence of these URLs when they're unauthorized could be considered a security breach in some scenarios.

It is recommended to only set it to `Admin Users Only` when troubleshooting, and then set it back to `None` when no longer needed.

4.6.59 Enable Spell Check

Syntax: select Yes or No button

This turns on the spell check option. With this option on, any search which produces no results displays a list of alternate-spelling queries, which will produce more results. If a query produces one result, Webinator suggests other words similar in spelling to the words you entered. The suggestions are based on the actual walk database, so unusual spellings or terminology used on your site are picked up by the spell-checker. The number of suggestions varies, depending on the `Suggest Time Limit` and `Number of Suggestions` options. The default is on.

4.6.60 Suggest Time Limit

Syntax: choose from drop-down list

This controls the number of seconds Webinator allows for spelling suggestions to be made. See also `Enable Spell Check` 4.6.59 for more information.

4.6.61 Number of Suggestions

Syntax: choose from drop-down list

This controls the number of spelling suggestions offered. See also `Enable Spell Check` 4.6.59 for more information.

4.6.62 Synonyms

Syntax: choose from drop-down list

This allows you to select a level of equivalence matching. You can limit results to specific matches, or you can allow synonyms and phrases. The values are described as follows:

`Disabled`: no phrase recognition and no synonyms (equivalences). Only searches for the the actual terms in a query. This is regardless of ~ usage.

`Phrase recognition only`: recognize query word groups that are known phrases and search for them as phrases.

`Phrases & Allow synonyms`: phrase recognition plus allows the tilde ( ) operator to match synonyms on specific query terms

`Phrases & Use synonyms by default`: phrase recognition and matching synonyms on all query terms (tilde to turn off on specific terms).

See also `Using the Thesaurus` (section 5.3).

4.6.63 Main Thesaurus

Syntax: the symbolic name for the primary thesaurus

Here you can select a main thesaurus. A drop-down list allows you to select one of the thesauruses that was defined in `System`, `Custom Thesaurus`.

See also `Using the Thesaurus` (section 5.3).

4.6.64 Secondary Thesaurus

Syntax: the symbolic name for the secondary thesaurus

Here you can select a secondary thesaurus. A drop-down list allows you to select one of the thesauruses that was defined in m`System`, `Custom Thesaurus`.

See also `Using the Thesaurus` (section 5.3).

4.6.65 Translate Boolean

Syntax: select Yes or No button

Off by default. If on, Boolean keywords `and`, `or`, and `not` in the search query will be translated into set logic.

Webinator uses set logic internally, and this setting translates basic boolean statements into proper set logic automatically. This is a limited translation, and does not support nesting of statements.

For more information on Webinator's use of set logic, please see the **Using Set Logic to Weight Search Items** of the Texis manual on our website.

4.6.66 Quotes for Literal

Syntax: select Yes or No button

Normally (when this setting is off, the default), double quotes around a group of search terms just makes the group a phrase: the terms must be found in the same order and adjacency as in the query. However, thesaurus lookups, word form processing (e.g. plurals), etc. are still performed on the phrase.

If **Quotes for Literal** is on, double-quoted terms will not only be phrases, but also searched for as is, without thesaurus lookups, word form processing, or other characters that would otherwise have a special query meaning.

4.6.67 Allow the @ Operator

Syntax: select Yes or No button

Off by default. If on, allow use of the `@` (intersections) operator in queries. Queries with few or no intersections (e.g. `@0`) may be slower, as they can generate a copious number of results.

4.6.68 Allow Linear

Syntax: select Yes or No button

Off by default. If on, an all-linear query –one without any indexable "anchor" words– is allowed. A query like "`/money #million`", where all the terms use unindexable pattern matchers (REX, NPM or XPM) is an example. Such a query requires a linear search of the entire table, and this can be very slow for a profile of significant size (e.g. 100,000+ documents).

If `allinear` is off, all queries must have at least one term that can be resolved with the Metamorph index, and a Metamorph index must exist on the field. Under such circumstances, other unindexable terms in the query can generally be resolved quickly, if the "anchor" term limits the linear search to a tiny fraction of the table. The error message "`Query would require linear search`" may be generated by linear queries if this is off.

Note that while enabling linear searches can improve the results of queries where it is needed, it also takes more time and machine resources – even more so that post-processing (p. 117). Thus careful thought should be given when considering enabling it, especially with large (e.g. 100,000+ document) profiles.

4.6.69 Allow "NOT" Logic

Syntax: select Yes or No button

On by default. If on, allows "not" logic (e.g. the − operator) in a query.

4.6.70 Allow Post-Processing

Syntax: select Yes or No button

Off by default. If on, post-processing of queries is allowed when needed after an index lookup, e.g. to resolve unindexable terms like REX expressions, or only partially indexable terms. If off, some queries are faster, but they may not be as accurate if they aren't completely resolved. The error message "`Query would require post-processing`" may be generated by such queries if this is off.

Note that while enabling post-processing can improve the results of queries where it is needed, it also takes more time and machine resources (though generally not as much as a full linear search). Thus careful thought should be given when considering enabling it, especially with large (e.g. 100,000+ document) profiles.

4.6.71 Allow Wildcards

Syntax: select Yes or No button

On by default. If on, wildcards are allowed in queries. Wildcards can slow searches somewhat because potentially many words must be looked for.

4.6.72 Allow Leading Wildcards

Syntax: select Yes or No button

Off by default. If on, leading wildcards ("`*word`") are allowed in queries. **Allow Wildcards** must also be enabled. Note that leading-wildcard terms are significantly slower to search for than trailing-wildcard terms such as "`word*`".

4.6.73 Single-Word Wildcards

Syntax: select Yes or No button

On by default. If on, wildcard searches will span only one word in the text – instead of up to 80 characters across words – and will suffix-match. E.g. the query "`con*tion`" will match "`condition`" but not "`consider my position`" nor "`conditionally`".

4.6.74 Allow WITHIN Operators

Syntax: select Yes or No button

Off by default. If on, "within" operators (`w/`) are allowed. These generally require a post-process to resolve, and therefore they can slow searches. If off, the error message "`'delimiters' not allowed in query`" will be generated if the within operator is used in a query.

4.6.75 Require All Words

Syntax: select Yes or No button

By default, all words a user searches for must be in the result for it to match. If `Require All Words` is changed to `N`, a result will be shown if *any* of the query terms are in the result.

Results that match multiple words will be ranked higher than results that match fewer.

4.6.76 Resolve Phrase Noise Words

Syntax: select Yes or No button

Off by default. This indicates whether to exactly resolve the noise words in phrases. If on, a phrase such as "`state of the art`" will only match those exact words; however, this may require post-processing to resolve (potentially slower). If off, any word is permitted in place of the noise words, and no post-processing is needed; this is faster but potentially less accurate.

4.6.77 Phrase Word Processing

Syntax: select box

This setting controls how suffix/wildcard processing – as determined by the **Word Forms** setting – is applied to phrases. Single-word terms always have suffix/wildcard processing applied; however phrases – multi-word terms bound by hyphens or in double-quotes – are only processed if this setting allows it.

There are two choices:

- `Last word only`: Only process the last word in the phrase; this is the default. For example, with this value set (and **Word Forms** set to `Any word forms`) the query "`"vacuum cleaner"`" would match "`vacuum cleaner`" as well as "`vacuum cleaners`".

- `None`: Do no processing on phrase words. With this value set, the query "`"vacuum cleaner"`" would only match "`vacuum cleaner`", regardless of **Word Forms**.

Note that a *single* word term is not a phrase – even if double-quoted – and thus **Phrase Word Processing** does not apply to it.

4.6.78 Keep Noise Words

Syntax: select `Y` or `N` button

Off (`N`) by default. This indicates whether to keep noise words ("`the`", "`and`", "`who`" etc.) in the query during query processing and search for them, or remove them from the query and ignore them. Searching for noise words can increase query time, as they occur frequently, and generally does not improve the results much due to their ubiquitous frequency; thus they are ignored by default.

4.6.79 Noise List

Syntax: whitespace separated list of noise (stop) words

A list of words to be ignored in queries (if **Keep Noise Words** is `N`). If empty the default list will be used, which is:

a	between	got	me	she	upon
about	but	gotten	mine	should	us
after	by	had	more	so	very
again	came	has	most	some	was
ago	can	have	much	somebody	we
all	cannot	having	my	someone	went
almost	come	he	myself	something	were
also	could	her	never	stand	what
always	did	here	no	such	whatever
am	do	him	none	sure	what's
an	does	his	not	take	when
and	doing	how	now	than	where
another	done	i	of	that	whether
any	down	if	off	the	which
anybody	each	in	on	their	while
anyhow	else	into	one	them	who
anyone	even	is	onto	then	whoever
anything	ever	isn't	or	there	whom
anyway	every	it	our	these	whose
are	everyone	just	ourselves	they	why
as	everything	last	out	this	will
at	for	least	over	those	with
away	from	left	per	through	within
back	front	less	put	till	without
be	get	let	putting	to	won't
became	getting	like	same	too	would
because	go	make	saw	two	wouldn't
been	goes	many	see	unless	yet
before	going	may	seen	until	you
being	gone	maybe	shall	up	your

4.6.80 Search Timeout

Syntax: integer number of seconds

This is the maximum overall time to spend searching and sending results. Exceeding this limit, whether due to server load, network slowness, etc. will result in a "`Timeout`" message to the user. This helps prevent heavy load from overwhelming the server. The default (if empty) is 30 seconds. The maximum is -1 for no limit, which is strongly discouraged.

4.6.81 Show Error Messages

Syntax: select box

Show Error Messages determines the disposition of error messages during searches. It may be set to one of the following values:

- `None`
  Suppress all errors
- `In HTML comments`
  Show errors in HTML comments (for HTML results styles) so that they are not normally visible to the user, but can be viewed via View Source in a browser. In XML results styles, errors will be suppressed.
- `In HTML comments & query errors visible`
  Show errors in HTML comments (for HTML results styles), but show query-related errors (e.g. "`Your query was all noise words.`") visibly (in gray boxes).

The default is `In HTML comments & query errors visible`. Note that in admin (test search) mode, all errors are always shown visibly, for admin perusal.

4.6.82 Debug SQL Level

Syntax: integer/hex number or empty/0 to disable

Setting Debug SQL Level to a non-empty/non-zero value (typically 3) enables extra debug messages for certain SQL statements. Generally only set at the request of tech support for diagnosing problems.

4.6.83 Debug Metamorph Level

Syntax: integer/hex number or empty/0 to disable

This enables extra debug messages for certain Metamorph statements. Generally only set at the request of tech support for diagnosing problems. Note that this setting can generate copious amounts of messages.

4.6.84 Search Trace Settings

Syntax: text

Debug/trace settings and values for searches, as a comma-separated list of "`param=value`" tuples. These are generally only set at the request of Thunderstone tech support, as they can cause copious tracing messages to appear, some of which may reveal authentication or other sensitive information. Supported settings and values are subject to possible change in future releases. See also **Walk Trace Settings**, p. 87, which has the same syntax.

4.6.85 Fast Result Counts

Syntax: select Yes or No button

Off by default. Some complex queries (e.g. those involving categories, or proximities closer than "page") can take more time to determine exact result hit counts. In some cases it may cause timeouts. Enabling this option will determine hit counts much faster, and using less CPU, though at the expense of accuracy. The hit

counts for complex queries will generally be overestimated (it will say there are more results than there really are).

4.6.86 Proximity

Syntax: choose a radio button

Proximity gives the ability to locate results with greater precision. The Webinator input form gives you several options to control the search proximity:

`line` - All query terms must occur on the same line

`sentence` - Query items must all reside within the same sentence

`paragraph` - Within the same paragraph or text block

`page` - All items must occur within same HTML document (the default)

Note that any value other than `page` requires post-processing – which can take some time – and thus **Allow Post-Processing** (p. 117) would need to be enabled. Thus if **Allow Post-Processing** is not enabled, the **Proximity** widget may not be shown in the search interface.

4.6.87 Language Characters

Syntax: list or range of characters, as inside REX `[]`

The **Language Characters** setting controls what characters constitute a language query. Query terms composed entirely of these characters are considered language terms, and have **Word Forms** processing applied. Additionally, during linear/post-process searches (e.g. hit highlighting on the Match Info page), potential matches of language or wildcard query terms will be expanded to include all adjacent characters that are part of this setting, and the match rejected if it does not match the query term (this prevents the query term `pond` from matching the text term `correspondence`, for example).

The syntax is a list of characters (no separation), and/or a range of characters; the same as a REX character class (without the brackets). The default is `\alpha\'\x80-\xFF`, i.e. alphabetic, hi-bit (for UTF-8) and apostrophe (for contractions). For best results, all characters that could match part of a **Word Definition** expression (p. 74) should usually also be listed in **Language Characters**.

See p. 193 for details on REX search syntax.

4.6.88 Word Forms

Syntax: choose from drop-down list

The `Word forms` options give you control over how many variations of your query terms are sought in your search as follows:

Exact match: Only exact matches are allowed. (the default)

Plurals & possessives: Plural and possessive forms are found. (`s`, `es`, `'s`)

Any word forms: As many word forms as can be derived are located.

Custom: use the three custom settings below to determine word forms.

4.6.89 Custom Suffix List

Syntax: Space-separated list of suffixes

When using the Word Forms `Custom`, this is the space-separated list of suffixes to use. All of these will be repeatedly stripped off of words, as long as the word is longer than the `Custom Suffix Min Length`.

An example setting could be `s es ' a e i y`. For the word `smith's`, the `s`and `'` would be stripped, causing it to match `smith`, `smiths`, etc.

4.6.90 Custom Suffix Default Removal

Syntax: Y or N

When using the Word Forms `Custom`, this controls whether to remove a trailing vowel, or one of a trailing double consonant pair, after normal suffix processing is finished. This will not apply if it would take the word below the minimum word length.

For example, if `ing` is in the suffix list and `Default Removal`is `Y`, then the word `running` will have the `ing` stripped, and then the 2nd `n` will be removed via `Default Removal`, producing `run`.

Default Removal is set to `Y` when using `Any Word Forms`, but not with `Plurals & Possessives`.

4.6.91 Custom Suffix Min Length

Syntax: Number

When using the Word Forms `Custom`, Webinator will not try to strip additional suffixes from any word shorter than this length. For example, if min length is `3`or more, the `es` on `yes` will not be treated as a suffix.

Min Length is set to `3` when using `Plurals & Possessives`, and `5` for `All Word Forms`.

4.6.92 Word Ordering

Syntax: choose from drop-down list

Controls how important word order is for results ranking: results with terms in the same order as the query are considered better. For example, if searching for "`bear arms`", then the hit "`arm bears`", while

matching both terms, is probably not as good as an in-order match. The default weight is `medium` (500). Note that search users can override this setting on the Advanced search form.

4.6.93 Word Proximity

Syntax: choose from drop down list

Controls how important proximity of terms is for results ranking. The closer the hit's terms are grouped together, the better the rank. The default weight is `medium` (500). Note that search users can override this setting on the Advanced search form.

4.6.94 Database Frequency

Syntax: choose from drop down list

Controls how important frequency in the table is for results ranking. The more a term occurs in the table being searched, the *worse* its rank. Terms that occur in many documents are usually less relevant than rare terms. For example, in a web-walk database the word "`HTML`" is likely to occur in most documents: it thus has little use in finding a specific document. The default weight is `medium` (500). Note that search users can override this setting on the Advanced search form.

4.6.95 Document Frequency

Syntax: choose from drop down list

Controls how important frequency in document is for results ranking. The more occurrences of a term in a document, the better its rank, up to a point. The default weight is `medium` (500). Note that search users can override this setting on the Advanced search form.

4.6.96 Position in Text

Syntax: choose from drop down list

Controls how important closeness to document start is for results ranking. Hits closer to the top of the document are considered better. The default weight is `medium` (500). Note that search users can override this setting on the Advanced search form.

4.6.97 Depth in Site

Syntax: choose from drop down list

Controls how important being close to one of the **Base URL(s)** is for results ranking. The more times the walk had to click on links to get to the document, the lower rank it will have. The default weight is `off`, i.e. do not factor in depth-in-site for results ranking. Note that search users can override this setting on the Advanced search form.

4.6.98 Date Bias

Syntax: group of drop downs and an optional date-picker

The **Date Bias** settings control how the relative date (age) of a document affects its result ranking. The older (or farther into the future) a document is, the lower its rank will be[1].

Weight is the importance of date for ranking, relative to other rank factors. It defaults to `off`, i.e. date will have no significance. Note that search users can override this setting on the Advanced search form.

Half-life is a measure of how fast the rank "decays" with document age. It is the time it takes for this rank factor to decrease to half of **Weight**. (The factor will reach 0 for an "infinitely" old document.) This can be tuned according to the profile's data set: an often-crawled news profile that has new articles appear hourly or daily, for example, might benefit from a **Half-life** of `1 day`, since its documents "age" over that time frame. On the other hand, a crawl of a company-wide document archive going back a decade or more might work best with a **Half-life** of `1 year` or even `5 years`. The default is `1 year`.

Field is the field to use for computing a document's age. It defaults to `Modified` (the document's `Last-Modified` date according to the server), but can also be set to `id` (the last time the crawl saw the document change). The **Field** chosen should also be set as one of the **Compound Index Fields** (p. 76) for best results; this will help ensure faster searching and more accurate result counts.

Anchor is the reference point or "best" date for the age of a document: documents with this date get the full **Weight** applied to their rank, while documents older or newer than this get less. It defaults to `Current Date`, i.e. right now.

Sometimes `Current Date` is not the best choice, however, because it is a moving target. For example, a daily crawl of news articles would see date biasing change throughout the day: an 8am article would rank higher when searched at 9am than when searched at 5pm. Setting **Anchor** to `Last Walk Finished` may help in this case: it uses the date of completion of the last successful walk – which will be fixed from search to search, yet still update with each walk.

In other cases, `Current Date` is not appropriate because the dataset is fixed. For example, a crawl of an unchanging historical archive from the 1990s – whose most recent document is from 1999 – should not see date biasing change for the same documents searched next year vs. now. Nor should it treat 1995 documents nearly the same as 1998 documents (because both are 20+ years old now). In this instance, it might help to set **Anchor** to `Fixed Date` and choose a date of e.g. `1999-12-31` in the **Fixed Date** date-picker that appears: this will treat 1995 documents as significantly older (4x) than 1998 documents.

4.6.99 Ranked Rows

Syntax: number

The maximum number of rows that can be scrolled to when returning ranked results. This can be set to 0 for all matching rows, or to any other number. The lower the number the better the performance, however users won't be able to scroll through as many results. The default is 200.

[1] Note: to order results by date *alone* – without regard to any other rank factors – simply set **Result Order** (p. 98) to `Date`. The **Date Bias** settings are for use when still ordering primarily by `Rank`.

4.6.100 XML Export Variables

Syntax: names separated by newlines

`XML Export Variables` is a list of variables, one per line, that are to be displayed and propagated through XML search results. This can be used to propagate HTTP headers from the client's request, or for including arbitrary extra information.

- `HTTP Headers`

  You can specify the header name in all caps, with underscores for spaces, and with a `HTTP_` prefix. For example, to include "User Agent", specify `HTTP_USER_AGENT`.

- `Arbitrary variables`

  You can specify any named variable, and values passed in to the search query will be propagated in the XML output.

For example, if you use `cbtGroup` and `HTTP_USER_AGENT`, and the search URL includes `...&cbtGroup=user&cbtGroup=backup&...`, then the following block will appear in the XML output, as a child node of `<ThunderstoneResults>`:

```
<exportVar>
  <variable name="HTTP_USER_AGENT">Mozilla/5.0 ....</variable>
  <variable name="cbtGroup">user</variable>
  <variable name="cbtGroup">backup</variable>
</exportVar>
```

4.6.101 Phishing Protection

Phishing Protection prevents Webinator from being used as a tool in a phishing attack.

Webinator has a redirect page as part of its Query Logging functionality, where it will provide a redirect to the URL specified. It would be possible for an attacker to specify a URL that, at first glance, looks like a link from Webinator, which the user may trust. After the redirect, it actually ends up somewhere else.

If Phishing Protection is enabled, the redirect page will make sure that any URL specified is actually in the profile's walk database before issuing the redirect to it.

4.6.102 Prevent Find Similar Fetch

This setting, when enabled, prevents Webinator from fetching URLs during custom "Find Similar" searches.

The `/similar.html` URL function (p. 142) can execute Find Similar searches with arbitrary URLs. The text of the URL is needed to formulate a Find Similar query. To find it, first the URL is looked up in the profile database. If it is not found, then only if **Prevent Find Similar Fetch** is `N` will the URL be fetched to find its text. By default, **Prevent Find Similar Fetch** is `Y`, so that users cannot cause Webinator to perform arbitrary URL fetches.

4.6.103 Decode Displayed URLs

Decode Displayed URLs will cause the URL that is displayed in search results to be URL-decoded, which includes replacing sequences with their proper characters.

This can be useful when URLs have words separated with spaces, which are replaced with `%20` to be a valid URL. Decode Displayed URLs allows you to display the decoded version, making the files easier for search users to read.

`"this%20is%20a%0file.txt"` becomes `"this is a file.txt"`.

4.6.104 Max Cache Entry Age

This sets the maximum age, in seconds, for a results cache entry. Entries older than this will not be used for results, and will be purged by the results cache manager. The default (if empty) is 21600 seconds, i.e. six hours.

4.6.105 Max Cache Size

This sets the maximum size, in bytes, of the results cache. This is not a hard limit, but when the cache grows larger than this, the results cache manager will start to remove old/low-priority entries. The default (if empty) is 100000000 (one hundred million bytes).

4.6.106 Min Search Time

This sets the minimum search time, in seconds, that a query must take in order to be considered for results caching. Queries that are faster than this are not cached, because they are considered fast enough to save the space. The value may be an integer or floating-point (decimal) number; the default (if empty) is 2.0.

4.6.107 Visible

This controls whether this profile is visible to other Webinators (or even the same one) for use in a meta search. Any profile that is to be used as a part of a meta search must have the `Visible` flag set to `Y`.

If a profile has `Visible` set to `N` and is used as a back-end for meta search, it will return the error `Profile not Visible`.

4.7 System Wide Settings

This area is for settings that affect Webinator as a whole and/or may be shared by multiple walk profiles.

4.7.1 Admin Theme

Selects a color theme to use for the Webinator's administrative interface. This doesn't affect walking or the search interface in any way.

It can be useful to provide a means of quick visual differentiation between multiple Webinators, or to change the contrast and/or color differentiation for accessibility reasons.

4.7.2 Admin Logo

This allows you to specify a custom logo to use in the upper-right hand corner of all the administrative interface pages.

This setting specifies full URL path that will be used in the `src` attribute of the image.

4.7.3 Default Profile

By default, accessing the `search` interface requires specifying a profile via the `pr` query variable. The Default Profile setting allows you to choose a profile that will be used if no profile is specified.

4.7.4 Cluster Members

This field defines the machine(s) and/or network(s) that constitute a cluster of Webinators. You can specify multiple addresses with a network prefix and wildcard (like `10.10.10.*`), netmask (like `10.10.1.0:255.255.255.0`), or address/prefixlen (like `10.10.1.0/24`) format. All machines matching these IPs will be allowed full access to Webinator internals without verification. This allows for replication and dataload.

If the request is forwarded such that a `X-Forwarded-For` header is included (like a load balancer), all IPs through the forwarding chain must be allowed by Cluster Members.

4.7.5 API Logging

Allows you to record the XML requests & responses of all dataload and SOAP admin API calls to `api.log` in the logs directory. This can be useful when troubleshooting why dataload requests aren't storing properly.

Dataload and replication are supported in the full Texis product, but not Webinator-only.

4.7.6 Task Monitor Logging

Controls the verbosity of logging for the Task Monitor. Messages are logged to `taskmonitor.log` in the logs directory.

4.7.7 Audit Logging

If enabled, all setting changes for all profiles and for **System Wide Settings** are written to the `logs/audit.log` file in the installation directory. This includes where the event came from, which account did it, which profile (if applicable), and what changed. Certain other events are included as well such as logins, logouts, failed logins, ACL-restricted events, profile creation/deletion, etc.

Note: While known sensitive fields (e.g. **Login Info**) have values redacted, other sensitive data may nonetheless be logged (e.g. URLs). See also the per-profile setting **Audit Logging** (section 4.5.88, p. 87).

4.7.8 Admin Banner

Sets the text to display at the top of every admin page when managing the Webinator. This can be used to display any information or warnings you want to constantly remind anyone when managing the Webinator.

4.7.9 Login Expiration

Allows you to customize the expiration of cookies served by the Webinator. This applies to logins to the administrative interface, and by default also applies to some results authorization logins (See 110).

Choose `Session` to provide login cookies that disappear when the user closes their browser.

Choose `Duration` to choose a custom duration for the cookies, defaults to `1 year`. Note that this duration is refreshed on every page load, so it acts as an idle timeout.

This setting only applies to cookies created by the Webinator and given to users of the Webinator. It has no effect on how cookies are handled when the Webinator acts as a client when walking websites.

4.7.10 Disable Starting All Walks

When this setting is on, no walks will launch for any profiles for any reason (manual, schedule, etc). Setting to `Y` will stop ALL profiles from walking, overriding any individual profile's `Disable Starting Walks` setting.

This can be useful with machines that should be dataload-only, or for machines that want to guarantee their content won't change.

Walks that are already running when this is set will finish normally.

4.7.11 Profile Dataspace Roots

This lets you force all profiles to be created beneath one or more specified directories. If any directories are entered, then at profile creation the `Data Directory` text input will be replaced with a selectable list of root directories, as specified here. The profile's data directory will automatically be created beneath the root directory based on the profile's name.

For example, if you set Profile Dataspace Roots to the single line `/usr/local/profileData` and you create a profile called `salesInfo`, its data directory will automatically be set to `/usr/local/profileData/salesInfo/`. If you set it to `E:\profileData`, the profile's data directory will be `E:\profileData\salesInfo\`.

4.7.12 Network Share Mounts Root

This setting is the root directory under which all network shares are expected to be manually mounted – locally on the Webinator machine – by the administrator, i.e. for `file://` URL walks. For example, if the value is set to `/mnt/local`, then the share `MyHost/MyShare` should be mounted on `/mnt/local/MyHost/MyShare`, for walking `file://MyHost/MyShare/...`.

This setting is ignored for profiles where **Network Share Access Method** (p. 90) is available, `Current`, *and* in effect; see details there. This setting is also ignored for profiles where **Proxy** (p. 82) is set, as the proxy is used instead. It also will not appear at all if Webinator is running under Windows (where UNC paths are used), or is not licensed for `file://` URLs.

4.7.13 System Replication Settings

"Targets" defines machines that will receive system replication from Webinator. This includes the creation and deletion of profiles, all profile settings changes, and all profile data. This also includes system-wide settings, thesauruses, and client certificates.

System replication targets must have `Allow Receiving` enabled, as described below.

Replication is supported in the full Texis product, but not Webinator-only.

4.7.14 Allow Receiving

Webinator can only be a receiver of System Replication (described above) if this Webinator has Allow Receiving set to `Y`. The sender must also be listed in this Webinator's Cluster Members, as described above.

4.7.15 Log All Replication

Writes information for each replication queue processor to `replication.log`. This forces logging for all profiles, and also for non-profile, System data replication.

If both "Log All Replication" and a profile's "Log Replication" are set, logging for that profile will be the more verbose of the two.

4.7.16 Experimental Features

This section may contain a list of experimental features in Webinator that can be enabled. Due to the experimental nature of the features there is no guarantee that they will be supported in future version, or that upgrades will work the same.

4.8 Results Authorization

Results Authorization allows restriction of search results to authorized users only, on a per-URL basis. Only users with access to a given URL will ever see that URL in a result list, instead of all users seeing all matches (and potentially being denied access to results already shown).

Access to a URL, as well as the namespace of users, is determined by the URL's origin server, not Webinator, so no reconfiguration of users or access is needed – the pre-existing server access controls are just forwarded by Webinator. And since access is determined on a per-result, not per-search, basis, a single profile can serve a multitude of users with any combination of whole/partial access to the underlying data.

Results Authorization works at search time (late binding) by accessing each potential search result URL with the user's credentials. Only URLs authorized to that user are then shown in search results. The authentication method(s) used will depend on the existing system(s) already used by the indexed URLs. Various schemes are supported:

- `None`: No access verification; return all search results to all users. This is the default.
- `Cookie-based`: Custom HTML-form-based single-sign-on systems. Users first login on a web server (not a Windows workstation login), which then sends an access cookie to the user's browser. This cookie is automatically returned to the server when accessing future pages, and grants the user access.
- `Basic`: HTTP Basic authentication, for web servers.
- `NTLM`: Windows NTLM authentication, for web servers.

- `SMB/Windows`: SMB for Windows file servers (for Thunderstone products that support `file://` walking).

For cookie-based systems, Webinator will merely forward the cookies the user has already received from the site login page. For all others (Basic/NTLM/SMB), Webinator must prompt for the user and password directly, as they are needed to verify result URLs. In the latter case, credentials will then be stored in a cookie by Webinator so that future searches do not need to re-prompt for a login. Note that NFS-mounted file servers are not currently supported by Results Authorization, due to limitations of NFS.

4.8.1 Results Authorization Walk Settings

Webinator itself needs read access to the entire set of URLs in order to build a search index. Therefore, before walking a protected data set for Results Authorization, it may be necessary to fill out the `Login Info` setting (p. 81) under `All Walk Settings` with a full-access admin type account, so that Webinator can walk the data.

Or it may be necessary to fill out a `Primer URL` (p. 78) containing login info to submit to a site's login form, so that Webinator can obtain the login cookies needed for access to the rest of the site.

4.8.2 Results Authorization Search Settings

After a successful walk, Results Authorization is configured with the `Results Authorization Options` group on the `Search Settings` page. The primary setting is `Authorization Method` (p. 108), which is determined by the authentication system(s) in use by the indexed URLs. If cookie-based, this is set to `Forward login cookies`; for all other systems, it is set to `Basic/NTLM/file - Prompt via form`. Most of the remaining settings depend on which method was selected; see the `Authorization Method` setting (p. 108) for details.

There are also a few resource/tuning settings, such as `Max Docs to Auth-Check`, `Successful Auth Result Limit`, `Total Auth Timeout`, and `Debug Results Authorization`, which are not required, but merely fine-tune the results.

4.9 Meta Search - Search multiple profiles as one

Meta search allows you to search multiple profiles simultaneously and merge and display the results as if it was one big profile. The meta search can search and combine profiles from multiple Webinators.

4.9.1 Profile Creation

When creating a profile, change the `Standard` select box to `Meta Search` instead.

4.9.2 Meta Search Walk Settings

`Walk Settings` is somewhat of a misnomer for a meta search profile since it doesn't do any walking of its own. On this page you list the host(s) and profile(s) to search and merge when this profile is accessed.

For each profile you want included in the search, list the full URL to that machine's `.../search` interface, e.g. e.g. `http://searchbox/texis/webinator/search/`. You can use `localhost` or `127.0.0.1` instead of the hostname when using profiles on the same machine as the meta search.

The **Display Name** column is used to provide a user friendly name for this profile that will be displayed if the user is allowed to choose which profiles to search.

The **Bias** setting allows you to apply a ranking bias to your metasearch targets. You can add or remove rank to results from a given target by increasing or decreasing the bias for that target. Setting bias to `3` will cause results from that target to have 3% higher rank than it normally would (a 76% result would become 79%, etc.).

The **Status** column shows the status of the remote profile once a host/profile has been entered and `Update` has been pressed. If the target is searchable, `OK` is displayed. Otherwise, text explaining the error is displayed. Refreshing the page re-queries the target profiles.

If **User Selection** is set to `Y` then the user will be presented with a list of **Display Name**s and can choose which ones to search. Leaving them all unchecked will cause them all to be searched. The list is submitted via the `mu` query string variable (one profile per `mu` value, multiple values if needed). If **User Selection** is `N` then any `mu` value(s) are ignored.

The **Meta Mode** setting controls whether profiles on the same host will be searched serially or in parallel. "Sameness" of host is determined by the **Target Search URL** setting, so using different names or a name and an IP address will allow you to mix serial and parallel.

The **Results Merge Method** setting controls how target profiles' results are merged and sorted by the meta profile. Two methods are available:

- `Requested order`
  The results will be sorted as requested, i.e. as specified by the `order` query-string variable (or if that is unset, the meta search **Result Order** search setting). This is the default. Thus, results from different target profiles may or may not be mixed together (depending on how they sort by `order`).

- `Target profile order`
  The results will be sorted by their **Profiles** setting order first, then by requested (`order` variable) order. This will result in *all* results of the first target profile being shown first, then all results of the next target profile, etc.

Note that in both cases, the *target* profiles still individually sort their results according to requested `order`. The **Results Merge Method** setting only affects how those top results are then merged and sorted by the *meta* profile.

Max Backend Data Size provides a sanity limit on the information collected from metasearch backends. You can usually leave this untouched, but it may need increased if you're getting `Max Page Size`

`exceeded` errors in your metasearch results. This can be caused by having an exceptionally large number of results per page (thousands), or very large Additional Fields.

4.9.3 Search Settings

The appearance options control the appearance of the meta search results pages. Currently the Results Authorization and query options of the meta profile do not apply: use the target profiles' options instead.

When using best bets the meta search profile must have the same group names as the backend profiles. Any best bets from the backends that have group names that are not defined in the meta profile will not be shown.

Query logging of the meta search and the backends are independent of each other. The meta search will respect its own query logging setting as will each of the backend profiles. So it is possible to have multiple logs for the same query if both the meta search and the backend have query logging turned on.

4.10 Access Control

Access Control allows different administrative users to be given different levels of access to Webinator; normally, with access control off (the default) all users have access to all administrative functions. Access Control can only be enabled or disabled by the `webinator` user.

Access Control is supported in the full Texis product, but not Webinator-only.

4.10.1 User Groups

User groups allow easier access control maintenance, as users with similar permissions can be administered together once rather than separately several times. The special group `Everyone` always exists and cannot be edited; it always contains all users as a convenience.

User groups may contain other groups as well as users, allowing complex hierarchies to be created if needed. Permissions for a user are affected by all groups a user is directly or indirectly a member of. For example, if user `Amy` is in group `Programmers`, and group `Programmers` is in group `IT`, then `Amy` is also indirectly a member of `IT`, and her permissions are affected by those granted to not only herself and `Programmers` but `IT` as well.

4.10.2 Object hierarchy

Each administrative action that can be access-controlled (e.g. editing walk settings, creating accounts) can be thought of as an object. Some actions are broader than others and can be thought of as a superset, e.g. editing *all* profiles is a superset of editing a *specific* profile. Thus, access control objects are arranged in a tree-like hierarchy, where each object has a parent object, and can inherit permissions from it. This makes setting privileges on a logical group of objects (e.g. all profiles) easier, as only one object may need to be changed (the parent). Also, when new child members (e.g. new profiles) are created, they will inherit the

same privileges automatically. The access control object hierarchy in Webinator is as follows:

```
/                       Global root object
   Users/                  User accounts
      webinator                   webinator user
      ...                      other users
   Groups/                 User groups
   Profiles/               Profiles
      default                  default profile
      ...                      other profiles
   Settings/               Profile settings
   Maintenance/            System page
      Info/
      Updates/
```

Note that these "files" do not really exist: the objects are merely symbols representing actions that can be access-controlled.

4.10.3 Access Control Lists

An object may have an Access Control List (ACL) associated with it. ACLs determine what rights (Read/Write/Delete/Change perms) users have on objects. Each object's ACL contains one or more Access Control Entries (ACEs). An ACE identifies a trustee (a user or group), a set of rights, and whether those rights are allowed or denied the trustee on that object. In addition to the ACL explicitly set on an object, rights may be inherited from parent objects' ACLs, as mentioned above.

4.10.4 Determining Effective Rights

The effective rights a specific user has on an object – what the user can actually do with the object – are determined by examining ACEs in a specific order. The first ACE that matches both the user and the desired access right determines whether the user has that right on the object. An ACE matches the user if it specifies the user or any group the user is directly or indirectly a member of. An ACE matches the desired right if the right is listed in the ACE.

ACEs are examined in the following order[2]:

1. ACEs explicitly set on the object
2. ACEs explicitly set on the object's parent
3. ACEs explicitly set on the object's further ancestors, nearest ancestor first

[2]In versions 5.3.0 and earlier, deny ACEs were always required to be before allow ACEs for an object.

At each object, ACEs are checked in ACL order (the order displayed for an object on the Access Control page). Order can be changed among multiple ACEs on the same object by using the `up arrow` and `down arrow` buttons next to the ACEs.

If no matching ACE is found after all levels are examined (back to the root or `Global` ACE), access is allowed by default (this is for back-compatibility with non-ACL mode).

4.10.5 Required Rights for Admin Actions

Certain ACL rights are required for certain administrative actions to be performed. In order to maximize rights-configuration flexibility, some actions require rights on multiple objects. For example, editing settings on a profile requires rights not only on the profile, but also on the setting itself. Note in the object hierarchy (p. 135) that profiles and settings are two "sibling" branches, rather than settings being replicated as descendants of every profile. Thus, profiles and settings can be thought of as a two-dimensional grid for permissions, and a user's rights can be tailored across that grid: access to one setting across all profiles, access to all settings on one profile only, etc.

The rights needed for specific actions are listed below. If a user does not have all of the required rights for an action, either a red `Access denied` message will be displayed, or (if access still granted to other parts) the affected object may simply not appear (read access denied), or may appear grayed out (write access denied). For more information and some example permission schemes, see the Using Access Control section, p. 164.

Walk and Search Settings

For settings under Basic, All Walk, and Search Settings, a user must have read access to the profile as well as read access to the specific setting in order to see the setting. Write access to the profile, and write and delete access to the setting, is needed in order to modify a setting. (Delete is needed to clear a setting, which may not be apparent from the form.) Note that some settings are grouped on a line, such as the `Robots` setting: permissions can be granted to the group as a whole (`Robots`), or only specific settings in the group (`Robots - robots.txt` or `Robots - Meta`). If a user has no read access to a setting, it will not be displayed on the page. If a user has no write access to a setting, it will be disabled (grayed out and not modifiable).

Starting and stopping a walk

Write access to the profile and write access to the `Walk now` setting is required to start a walk. Write access to the profile and write access to the `Stop walk` setting is required to stop a walk.

Best Bets

Write access to the profile and write access to the `Best Bet Groups` setting is needed to modify the Best Bet Groups for a profile, or to modify Best Bet words for a specific URL (under List/Edit URLs). Note

that this is distinct from editing Best Bet *search* settings (e.g. `Top Best Bet Title`), which only affect search, not the walk itself.

List/Edit URLs

Write access to the profile and write access to the `List/Edit URLs` setting is needed to modify URLs in the database, including using the `Update Soon` link. Read access to both is needed to view URLs.

List Duplicates

Read access to the profile and read access to the `List Duplicates` setting is needed read the error table and list the duplicates of a URL.

Walk Status

Read access to the profile and read access to the `Walk status` setting is needed to view Walk Status.

Query Log

Read access to the profile and read access to the `Query log` setting is needed to view the Query Log.

Profiles

Read access to the profile and read access to the desired setting(s) are needed to view the given setting. Write access to both is needed to modify a setting. Delete access to the profile is needed to delete the profile. Write access to `All Profiles` (the parent of profiles) is needed to create a new profile.

Accounts

Write access to `All Users` is needed to create a new user. Write access to the user is needed to change the password for a user. Delete access to the user is needed to delete a user.

User Groups

Write access to `All Groups` is needed to create a new group. Write access to the group, as well as write access to each member being added or removed, is needed to add or remove members to or from a group (except where the group is only indirectly being modified due to a member itself being deleted). Delete access to the group is needed to delete a group.

Access Control

Change-perms access to an object is needed in order to create, edit or delete an ACE on the object.

Maintenance

The following `Maintenance` objects control access to various system resources:

- `Info`: Read access is needed to access the `System` → `Information` → `System Information` menu and its links, and the `Dashboard`.

  Read access to `Maintenance/Info/TestNetwork` is needed to access the **System** → `Information` → `Test Network and Servers` page. (Alternatively, it can also be accessed with profile and `Settings/TestFetch` read access, since the **Test Page Fetches** form may use profile settings and the page is also reachable via **Tools** → `Test Fetch`. However this will not grant access to the **Test Network** form.)

- `Updates`: Write access is needed to install or update the software, or to apply a license via the GUI.

4.11 Running the Walker by Hand

4.11.1 Using dowalk

Normally a walk is initiated from the administrative interface. There may, however, be times when it is desirable to start a walk by hand from a shell (or command) prompt or as a part of some other automated task. When the administrative interface starts a walk it shows you the command line to use It is of the form:

```
texis profile=PROFILENAME dowalk/dispatch.txt
```

You may also specify the parameter `ttyverbose` to be 1, or higher, to tell `dowalk` to print various status messages to the screen when being run by hand. The form would be

```
texis profile=PROFILENAME ttyverbose=1 dowalk/dispatch.txt
```

Where `PROFILENAME` is the name of the profile you have configured using the administrative interface. You will need to supply the full path to texis if it is not in your PATH. You will also need to supply the path to the `dowalk` script if it is not in the current directory when you run the command.

INSTALLDIR`/bin/texis profile=`*PROFILENAME*`□↪`
`↪`*INSTALLDIR*`/texis/scripts/webinator/dowalk/dispatch.txt`

or

INSTALLDIR`\texis profile=`*PROFILENAME*`□↪`
`↪`*INSTALLDIR*`\Texis\Scripts\Webinator\dowalk/dispatch.txt`

The walker will behave the same as it does from the administrative interface. Walk info will be logged to the same files. See section 6.4.

There are several other "entry points" that can be used to get various different behaviors when starting the walker. They all take the same form as `dispatch` above except that `dispatch` is replaced by the name of the entry point. The entry points are:

- `dispatch`
  Starts a walk on the profile, using the profile's `Rewalk Type` setting to determine if a New or Refresh walk should be performed.

- `hold`
  Stops a walk that is in progress, create/update the search indices and make it the live search.

- `stop`
  Stops and abandons a walk that is in progress.

- `indexmakelive`
  Creates/updates the search indices on an abandoned walk and makes it the live search.

- `refreshnow`
  Mark a URL for refresh-ASAP. This requires an extra `u=THEURL` argument to tell it what URL to refresh. This will flag the page for refresh on the next refresh walk. It will not refresh anything itself. So you need to have walk type set to refresh and a schedule set.
  `texis profile=PROFILENAME u=THEURL dowalk/refreshnow.txt`

- `ifmodified`
  Checks the `Watch URL`. If the watched page has changed a walk is started. If not no action is taken. This is generally used on a frequent schedule to automatically rewalk a site if it changes.

- `singles`
  Fetches and indexes any single pages specified in the profile that are not yet in the database. You would call this after adding adding to `Single Page`, `Page File`, or `Page URL`.

- `recat`
  Recategorizes the database based on the current settings of `Categories`. This may take some time on large walks.

- `updateindex`
  Updates the Metamorph index on the html table. This would be used after performing manual SQL operations against the html table.

- `remakeindex`
  Drops and recreates all (standard) indices on the database. This has little use except in the case where indices are corrupted by disk errors or such.

- `remakemmindex`
  Drops and recreates the Metamorph index on the html table. This would be used after changing the `Word Definition` expressions.

- `tsverrors`
  Dumps the error table as tab separated values of `Date`, `Url`, `Reason`. Optional `start` and `end` date-times may be specified. Not specifying `start` means start at beginning. Not specifying `end` means continue to end. `texis profile=`*PROFILENAME* `start="2004-10-01"`□↪
  ↪`end="2004-11-01" dowalk/refreshnow.txt`

4.12 Running the Search Interface

See section 5.1, p. 141.

Chapter 5

Procedures and Examples

5.1 Searching your Index

Search the pages you have indexed by entering the following URL into your Web browser:

- On Unix:
  `http://www.example.com/cgi-bin/texis/webinator/search/`
- On Windows using CGI:
  `http://www.example.com/scripts/texis.exe/webinator/search/`
- On Windows using ISAPI:
  `http://www.example.com/texis/webinator/search/`

The above is a virtual path comprised of 2 parts. "`.../cgi-bin/texis`" is the Texis Web Script interpreter and "`/webinator/search`" is the path to the search script relative to your installation's `ScriptRoot`, which is the `texis/scripts` subdir of your install dir.

You may have to use a slightly different URL if you specified a different CGI directory during installation.

The URL given above will search the live database specified in the default profile called "`default`". If that profile is not found it will try to search the default walk database, $INSTALLDIR$`/texis/db` on Unix or $INSTALLDIR$`\texis\db` on Windows.

You may specify an alternate profile by including its name in the URL.

```
.../webinator/search/?pr=MYPROFILE
```

Where `MYPROFILE` is the name of the profile you wish to use. The search will use the live database specified by that profile.

You may also specify a database to search instead of a profile.

```
.../webinator/search/?db=DATABASE
```

Where DATABASE is the name of the database you wish to use. This would generally be the live database for a given profile which may be found as the first item listed on the administrative interface's Walk Settings page. Databases used this way must exist under the texis subdirectory of the installation directory. What you specify for DATABASE is only the portion of the path and name under the texis directory. For example, to search the database /usr/local/morph3/texis/myprofile/db2 you would use:

```
.../webinator/search/?db=myprofile/db2
```

When using a database instead of a profile, the look and feel settings will be those that were live when the walk of that database was performed. The profile will not be consulted for more recent changes. A benefit of not consulting the profile, however, is some increased search speed, which may be useful on a very heavily searched system. A disadvantage of specifying the database is that it will no longer be correct if a new walk is performed.

To get help on constructing queries click on the Advanced button of the search form. On the advanced search form you will find hyperlinks into the search help, which is also included in this manual in section 7.

To place the search form onto your existing web page(s) call up the Live Search from the administrative interface main menu (or the URL you determined from the above). This will bring up the search form. Use your web browser's view page source option (MSIE: TopMenu → View → Source) to get the source of the page. Cut everything between and including the <FORM> and </FORM> tags. That form may then be pasted into the web page(s) of your choice. You may also rearrange the look of the form as long as the variables are still present. If you have categories there will be a category select list in the form. You may leave this out if you always want to search everything. Or you may make it a hidden variable with a fixed value if you always want to search the same section.

5.2 Similarity Searching

The search script has a feature called "Find Similar" which gives a link for each result record that, when clicked on, finds more pages within the database similar to that result. This feature may also be accessed from any web page by placing the appropriate URL on it. You may search for pages in your database that are similar to any other web page whether it's in the database or not. The URL for finding similar pages has the form shown below.

Note: On Windows the /cgi-bin/texis/ *portion of the following URLs will be something like* /scripts/texis.exe/ *but may vary depending upon your installation.*

```
http://www.example.com/cgi-bin/texis/webinator/search/↪
   ↪similar.html?pr=default&ref=http://example.com/somepage.html
```

If the page containing the similarity URL resides on the same server as the search the http://www.example.com portion may be omitted:

```
/cgi-bin/texis/webinator/search/similar.html?↪
   ↪

pr=default&ref=http://example.com/somepage.html
```

If the profile to be searched is "default" the `pr=default&` portion may be omitted:

```
/cgi-bin/texis/webinator/search/similar.html?↪
   ↪ref=http://example.com/somepage.html
```

If the profile to be searched is anything other than "default" that must be specified instead of `default`:

```
/cgi-bin/texis/webinator/search/similar.html?↪
   ↪pr=myprofile&ref=http://example.com/somepage.html
```

If the page to be located is the page the URL is on the `ref=URL` portion may be omitted, as the HTTP `Referer` is used by default:

```
/cgi-bin/texis/webinator/search/similar.html
```

or

```
/cgi-bin/texis/webinator/search/similar.html?pr=myprofile
```

The similar function will look up the desired URL in the database or, if it's not in the database, fetch it from the webserver (if **Prevent Find Similar Fetch** is disabled, p. 126). It will then search the database looking for pages similar to the specified page.

You could place a URL like this on all of your pages so users could, with one click, find all pages on your site similar in content to the one they were reading.

5.3 Using the Thesaurus Feature

You can create a thesaurus to either replace or add to the default thesaurus. The creation procedure is the same for either usage. Note that a thesaurus is not limited to synonyms. It can contain anything you wish to associate with a particular word: i.e., identities, generalities, or specifics of the word entry, plus associated phrases, acronyms, or spelling variations. Webinator maintains a collection of thesauruses that you upload. For each profile you may select which, if any, thesaurus to use.

Here are the steps to use the thesaurus feature.

- Create a thesaurus file. Use the syntax described in the document "User Equivalence File Format" at the following URL: `http://docs.thunderstone.com/site/texisman/↪user_equivalence_file_format.html`

  That document refers to the thesaurus as an "equivalence file".

- Upload your thesaurus to Webinator. At the main menu click `System`, `Modules`, `Thesaurus`. Existing custom thesauruses can be downloaded by clicking on their name.

- In the `Name` field, enter a symbolic name that will be listed as an option in search settings. This name does not have to be related to the filename on disk in any way The name can contain letters, numbers, dash, or underscore.

- In the Permutations field, choose a value. This value controls how many variations of your defined terms to create during indexing of your uploaded source file. Here is an example of the effect of the various values.

  Assume a thesaurus entry of: `car,ford,chevy,toyota`

  `Permutation None`: Just the terms as you entered them. Query "car" would find "car", "ford", "chevy", and "toyota". Query "ford" would only find "ford".

  `Permutations Single`: The terms you entered and the reverse. Same as above plus a query for any of "ford", "chevy", or "toyota" would find "car".

  `Permutations Full`: Equate every term with every other in each entry. Same as above plus a query for "ford" would find "chevy" and "toyota".

- In the `New File` field, enter (or browse to) the file on your disk to upload. Click `Save Changes` to upload and index the file. When indexing is completed, you will receive a report about the indexing. If `Show results of indexing` is checked, you will also get a summary of the indexed words.

- After your thesaurus is installed on Webinator you can go to `Search Settings` for a profile to activate the thesaurus. There are three related options: `Synonyms`, `Main thesaurus`, and `Secondary Thesaurus`.

- Set `Synonyms` using the following information. `Synonyms` indicates how you want to apply a thesaurus (either yours or the default) to queries.

  `Disabled`: no phrase recognition and no synonyms (equivalences)

  `Phrase recognition only`: recognize query word groups that are known phrases and search for them as phrases

  `Phrases & Allow synonyms`: phrase recognition plus allowing the tilde ( ) operator to match synonyms on specific query terms

  `Phrases & Use synonyms by default`: phrase recognition and matching synonyms on all query terms (tilde to turn off on specific terms).

- Set the `Main Thesaurus` and `Secondary Thesaurus` fields by using the following information. If you want to use only your thesaurus and not the default one, select yours for the `Main Thesaurus` option and leave verb'Secondary Thesaurus' set to none. If you want the default in addition to your own, leave `Main Thesaurus` set to Built-In and set `Secondary Thesaurus` to yours. The names listed in these options are the symbolic names (`Name` field) you gave your thesauruses when uploading them.

- Click `Update` to apply these settings. There is no need to check `Apply Appearance`, and these settings are applied to both `Test Search` and `Live Search`.

5.4 Page Exclusion, Robots.txt, and Meta-robots

On the first access to a site the file `/robots.txt` will be retrieved, if it exists. Settings there will be respected. Any encountered URL that is disallowed by `robots.txt` will be discarded. Meta robots is

also respected for each page retrieved. See `http://www.robotstxt.org/wc/exclusion.html` for the robots.txt and meta robots standards.

If there are any HTML trees that you don't want indexed you may want to setup a `robots.txt` file, meta robots within the HTML pages, or use the various exclusion options to Webinator. For example: if you had a "text only" version of your web server that duplicated the content of your normal server you would not want to index it. (On the other hand if most of your meaningful text is contained in graphics, Java, or JavaScript you may want to walk the text tree instead of the normal one, since graphics and Java are not searchable.)

Suppose your "text only" pages were all under a directory called `/text`. The simplest way to prevent traversal of that tree would be to use the exclusion or exclusion prefix.

The exclusion would look something like this:

```
/text/
```

The exclusion prefix would look something like this:

```
http://www.example.com/text/
```

That will prevent retrieval of any pages under the `/text` tree. This does not prevent other Web robots from retrieving the `/text` tree. To setup a permanent global exclusion list you need to create a file called `robots.txt` in your document root directory. The format of that file is as follows:

```
User-agent: *
Disallow: /text
```

Where "`*`" is the name of the robot to block. "`*`" means any robot not specifically named (all robots in this case since no others are named). Or you could specify the name of the robot. For Webinator it would be `Webinator`. You may specify several "Disallow"s for any given robot (see below). The "Disallow"s are simple path prefixes. They may not contain wildcards.

You may also specify different "Disallow" sets for different robots. Simply insert a blank line and add another "User-agent" line followed by its "Disallow" lines.

Here's a larger example:

```
User-agent: *
Disallow: /text
Disallow: /junk

User-agent: Webinator
Disallow: /text
Disallow: /webinator

User-agent: Scooter
Disallow: /text
Disallow: /junk
Disallow: /big
```

The `Scooter` robot will be blocked from accessing any pages under the `/text`, `/junk`, and `/big` trees. Webinator will be blocked from accessing any pages under `/text` and `/webinator`. All other robots will be blocked from accessing pages under `/text` and `/junk`.

Use of `robots.txt` is not enforced in any way. Robots may or may not use it. Webinator will, by default, always look for it and use it if present. This may be disabled by turning off **robots.txt** under the **Robots** setting. When using robots.txt you may still use "Exclusions" for manual exclusion.

Meta robots provides another method of controlling robots such as Webinator. Any HTML may contain a meta tag in the source of the form.

```
<meta name="robots" content="WHAT-TO-DO">
```

`WHAT-TO-DO` may contain any of the following keywords. Multiple keywords may be used by placing a comma(`,`) between them.

Table 5.1: Meta-Robots Flags

Keyword	Meaning
`INDEX`	Index the text of this page
`NOINDEX`	Don't index the text of this page
`FOLLOW`	Follow hyperlinks on this page
`NOFOLLOW`	Don't follow hyperlinks on this page
`ALL`	Synonym for `INDEX,FOLLOW`
`NONE`	Synonym for `NOINDEX,NOFOLLOW`

Like `robots.txt` this is not enforced in any way. Robots may or may not use it. Webinator always indexes and follows hyperlinks by default so it only looks for `NOINDEX` and/or `NOFOLLOW` and/or `NONE`.

5.5 Indexing Other Sites

You may index a site other than your own by specifying its URL just as you would for your own site.

```
http://www.anothersite.example.com
```

Please be kind when indexing other sites. Many are low bandwidth or heavily used already and won't appreciate being hit hard. If you want to index any significant number of sites, please contact Thunderstone, as we may have what you want already.

5.6 Indexing Individual Pages

To add an individual HTML page to the database, but not go after any of its references, add it to the `Single Page` list box.

5.7 Reindexing on a Schedule

It is often desirable to reindex a given site on a regular basis because of continuously changing content. You may specify a `Rewalk Schedule` to handle this for you.

It is also useful to perform a single rewalk at a later time or date to avoid overloading a web server during heavy use periods.

5.8 Checking for Web Server Errors

When you start a walk you will be sent to the walk status page. You may also reach that page at any time by selecting `Walk Status` from the menu. This page will show you the summary status of the running walk. When the walk completes you will see a summary of the walk as well as a list of any errors encountered. Following the error list is a list of duplicate pages encountered.

You may also view document linkage and info and errors from the `List/Edit URLs` page (4.3) from the menu.

5.9 Removing Pages from the Database

Use the `List/Edit URLs` menu (4.3) to find and delete specific URLs from the the database. You may delete individual pages or many pages at once using wildcards.

5.10 Troubleshooting missing content URLs

"Why didn't this content get indexed?" is a common first troubleshooting problem.

The first step is determining a specific content URL that you would expect to be part of the searchable content, but isn't. We'll refer to this as the "Content URL".

- Use the `Tools` → `List/Edit URLs` interface too look up the Content URL. Is it present in the searchable database?
  - If so, clicking on the listed URL to go to the List/Edit Details page for the Content URL. Here you can compare its content to what you expect, and view any errors.
- If the Content URL isn't in the index, you need to determine a URL that links to that Content URL (we'll call this "Parent URL").

  Now look up the Parent URL in `Tools` → `List/Edit URLs`. Is the Parent URL in the index?
  - If not, we need to repeat the process again, thinking of a URL that links to THAT Parent URL, and try looking that one up, until you find one that IS in the index. We need to find the break in the "chain" of links between your Base URL, and the Content URL.
- With the Parent URL found in the index, click on it to see its List/Edit Details page. On the Details page, click `Children` link to see what links were found on that page and see if the missing page is listed.

  Is the missing link among the listed Children links, and is there an error next to it?
  - If it's not there at all, Webinator might not be processing your Parent URL correctly, please get in touch with Thunderstone Support.
  - If it's listed and there's a error, that should describe why it's not present.
  - If the URL is there without an error, then Webinator chose not to index the URL because of some rule, such as robots.txt, meta robots, exclusions, max pages, max depth, exclude by field links, etc. Walking again with a higher `Verbosity` value, such as `4`, may help explain why it wasn't walked.

5.11 Erasing the Entire Database

If you decide to wipe out your existing database and its settings to start over go to "Profiles" and click "Delete" next to the profile you wish to delete. This will completely remove the selected walk database and all options related to it.

5.12 Using Multiple Databases

Once you have a live searchable database you may want to build a separate one to contain different kinds of pages or to experiment with, without destroying your live database. Use the `Profiles` menu to create a

new profile and database. You create the new profile with default settings or with a copy of the settings from another profile.

5.13 Integrating Search with your Site

There are four main techniques to integrate Webinator with your site. The techniques are grouped as follows:

- Link to the Webinator
- Embed a search box
- Request XML search results
- Invoke the search SOAP API

The first two are simple to implement. They involve only static HTML content, and can be used in situations where no dynamic scripting is available.

The latter two are more powerful and can be used in dynamically generated web sites. You'll create a "search page" for your site (search.php, search.aspx, etc) which accepts a query from the search user. Your search page makes a search query (HTTP request) to Webinator, and receives result data. Your search page then displays the data in whatever manner you wish to the search user.

The advantage of the latter two methods is your site controls all interaction with the search user, making it easy for your search results page to "inherit" the look and feel of your site. They never contact Webinator directly.

5.13.1 Link to the Webinator

When you want to make a HTML link to your profile's search interface, select the appropriate profile in the admin interface and click `Search` in the menu bar.

The address of that search page can be used in a link to search that profile, such as:

`<a href="`*searchAddress*`">Search</a>`

Where *searchAddress* is the URL of the search page.

5.13.2 Embed a search box

It's possible to have a small search box on your pages where the user can type a query and submit searches. The search box submits to the Webinator and users will have the search results served by the Webinator, although the look and feel of the search results page can be customized to look like your site.

To acquire the HTML needed for an embedded search box:

- In the admin interface, click `Profiles` in the menu bar and select the profile you'd like to use in the search box.
- Click `Search` in the menu bar. This opens the search form.
- Use your web browser's "View Page Source" option (`View` → `Source` in the menu, or `Ctrl+U`) to open a window that contains the HTML source code of the page.
- Copy everything between and including the `<form>` and `</form>`.
- Paste the form into your web page(s).
- Add the Webinator's hostname to the beginning of the form's `action` attribute. For example, if your Webinator is `search.example.com`, and the existing attribute looked like:

  ```
  <form action="/texis/...
  ```

  You'd change it to:

  ```
  <form action="http://search.example.com/texis...
  ```

5.13.3 Request XML search results

Your own dynamic php/asp.net/etc pages can issue a query to Webinator, and receive back XML results.

Issuing a Query Programmatically

Here is an example URL for a simple XML search:

`http://`*HOSTNAME*`/texis/webinator/search/main.xml?pr=`*profile*`&query=`*query*

Where *HOSTNAME* is the IP/hostname of your Webinator, *profile* is the profile to search, and *query* is the user's query. The URL path for your installation may be different from `/texis/webinator/search`, use whatever normally appears in your search URLs.

Search Parameters

The possible parameters that can be used in the query string are:

Queries:

- `category` - Category to limit results to, specified by name. Can be provided multiple times, or as pipe- ("`|`") separated values list, to limit results to those with any of the specified categories. Added in version 20.1. See **Categories**, p. 55.
- `requireAllCategories` - Setting to `Y` requires each result to be in *all* specified categories, instead of any one of them. Added in version 20.1. See **Categories**, p. 55.
- `cq` - Category to limit results to, specified by number: `cq=1` for first category, `cq=2` for second, etc. Can be provided multiple times or as a CSV to limit results to those with any of the specified categories. See **Categories**, p. 55. Deprecated; use `category` instead.

- `cqall` - Setting to `Y` requires each result to be in *all* specified categories, instead of any one of them. See **Categories**, p. 55. Deprecated; use `requireAllCategories` instead.
- `dq` - Maximum Depth query (e.g. `dq=2` for results found at most 2 links away from **Base URLs**)
- `mtq` - Mime Type query. May be an exact literal (e.g. `mtq=application/pdf`) or have a `*` after the `/` for anything of the left type (e.g. `mtq=text/*`)
- `query` - Main search query
- `sq` - Site query (p. 102)
- `tq` - Title-only query (same Metamorph syntax as keyword search)
- `uq` - URL query, match against the entire URL. Accepts wildcards `*` for any amount of anything and `?` for any single character. (e.g. `uq=https://www.example.com/dir/*`)

Search control:

- `dateSource` - What date to use, `id` or `Modified` (see below)
- `mdgt` - Modified Date Greater Than: Only results with a modified date less than this will be returned.
- `mdlt` - Modified Date Less Than: Only results with a modified date less than this will be returned.
- `pr` - Specifies the Webinator profile
- `prox` - Proximity: Only return results with the query words in the same `line`, `sentence`, `paragraph`, or `page` (default). Sets the **Proximity** search option (p. 122).
- `order` - Controls the sort order; this is the same variable that **Result Order** (p. 98) controls. Valid values are `r` (relevance), `dd` (date descending: newest first), or `da` (date ascending: oldest first). The date used for newest/oldest sorting is the `Last-Modified` date of the document (or date of walk if none); this can be changed via the `dateSource` parameter (p. 153). In version 17.1 and later, `rank` may be given as an alias for `r` (relevance), `date` for `dd` (newest first), and `size`, `visited` or `depth` may also be given. `order` may also be an Additional Field to sort by; see Sorting under Additional Fields, p. 177.
- `rpp` - Max number of results per page (up to permitted limit; see **Max User Results Per Page** search setting, p. 101)
- `sr` - Max number of results per site (if permitted; see **Results per Site** search settings, p. 101)
- `sufs` - Word forms (suffixes). Values are `0` (Exact match), `1` (Plurals & Possessives), `2` (Any word forms), or `3` (Custom)
- `mu` - For meta searches: each value is a target profile (display name) to search. Can be used to narrow down target profile search list. Only respected if **User Selection** (p. 133) is `Y`.

Rank Knobs: control the influence of ranking factors. Unless otherwise specified, each is an integer value, from off (`0`) to maximum (`1000`), to indicate the relative weight of that factor. Medium (`500`) is the default.

- `rorder` - Word ordering: Favors results with query terms in the same order as the query; overrides **Word Ordering** (p. 123)
- `rprox` - Word proximity: Favors results with query terms close together; overrides **Word Proximity** (p. 124)
- `rdfreq` Database frequency: Favors results with query terms more rare across the entire profile; overrides **Database Frequency** (p. 124)
- `rwfreq` - Document frequency: Favors results with query terms repeated more often in the document; overrides **Document Frequency** (p. 124)
- `rlead` - Position in text: Favors results with query terms earlier in the document; overrides **Position in Text** (p. 124)
- `rdepth` - Depth in site: "Shallower" results (fewer clicks from the Base URL) are better; overrides **Depth in Site** (p. 124)
- `rdatebiasWeight` - Date bias weight: Favors "newer" results (closer to `rdatebiasAnchor` i.e. `now`). Additional parameters:
  - `rdatebiasHalfLife` - Date bias decay rate: time for `rdatebiasWeight` to be halved, in seconds
  - `rdatebiasAnchor` - Date bias reference point: "best" date for maximum rank; can be `lastWalkFinished` for completion date of last successful walk, or Texis-parseable date
  - `rdatebiasField` - Date bias field: date field to use for computing document age (default `Modified`)

  All of these override the equivalent **Date Bias** value in Search Settings (p. 125).

Additional Fields: To add search restrictions to the query you can specify form variables with a name constructed as `af`nOP, where n is the number of the additional field (1, 2, or 3), and OP is one of the following operations:

- `eq` - the field is equal to the form variable (e.g. `af1eq`)
- `gt` - the field is greater than the form variable (e.g. `af2gt`)
- `gte` - the field is greater or equal to than the form variable
- `lt` - the field is less than the form variable
- `lte` - the field is less or equal to than the form variable
- `like` - the field matches the form variable (text search). This has the same syntax and functionality of the Metamorph query engine used in the main text search.

Examples:

- `af1eq=important` - only results where the first additional field is set to `important`

- af2lt=100 - only results where the 2nd additional field is less than 100.
- af3gte=2010-01-01 - only results where the 3rd additional field is 2010 or newer.
- af1like=important - only results where the first additional field contains the word important
- af1like=(critical,important) - only results where the first additional field contains either important or critical

dateSource: id vs modified

The dateSource parameter allows you to determine which date associated with the URL gets used for display, sorting, etc.

- Modified (default) - The time the content was last modified
- id - The time that Webinator last updated its record of the content

If a collection of files that were modified a year ago were picked up by the Webinator walk last night, then the Modified date would be a year ago, but the id date would be last night.

id is the default dateSource when requesting an RSS feed of a search.

Other Variables

- dropXSL - When **Results Style** (p. 99) is set to XSL Stylesheet, or the search request URL is search/main.xml (note the .xml extension), the dropXSL query variable controls how the XSL is applied. It may have one of the following values:
  - html - Apply the XSL style sheet server-side and serve the resulting HTML
  - no - Do not apply XSL, nor give an XSL reference. Browsers will display the raw XML. Can be useful for debugging/analyzing XML results.
  - yes - Do not apply the XSL, but give an XSL reference so that browsers will fetch and apply the XSL client-side.

  If not given, dropXSL is set based on the search/main.*ext* file extension in the URL: no if .xml was given, html otherwise.

XML Elements in Search Results

Search results can be sent as XML from Webinator to the host server. This section describes the XML elements.

<ThunderstoneResults> Overall container for the search results

- <XmlOutputVersion> Defines the version of this xml output
- <Query> Main text search string

- `<TitleQuery>` Query applied only to titles
- `<UrlQuery>` Query applied to URL
- `<DepthQuery>` Maximum Depth
- `<MimeTypeQuery>` Query applied to Mime Type
- `<CategoryQuery>` Numeric index of a category to require results to be in. `1` is the first category, etc. Deprecated; use `<CategoryName>` instead.
- `<CategoryName>` Name of a category to require results to be in. Added in version 20.1.
- `<RequireAllCategories>` Set to `Y` to only match results in *all* specified categories instead of any one (or more) of them.
- `<ResultsPerSiteQuery>` Max results per site, as specified by user
- `<UserResultsPerPage>` Max results per page, as/if specified by user
- `<TextQuery>` Text part of main search query
- `<TextQueryHighlight>` TextQuery with query highlighting (if enabled)
- `<PreviousRefine>` Additional refine queries
- `<SiteQuery>` Site query (from `site:` host in the query, or dedicated `sq` query string variable)
- `<LinkQuery>` Link query (from `link:` URL in the query)
- `<InFieldQueriesAllowed>` Set to `Y` if `infield:` queries (a Parametric search operator) are allowed
- `<ModifiedDateLessThan>` Only return results with Modified date earlier than this
- `<ModifiedDateGreaterThan>` Only return results with Modified date greater than this
- `<UrlRoot>` URL root of the search script, for making links
- `<Profile>` Profile used
- `<dropXSL>` Whether to apply or drop the XSL stylesheet
- `<AdvancedSearch>` Set to `1` if the advanced form should be displayed
- `<Proximity>` Proximity used for the search. Possible values:
  - `line` - Must occur on the same line
  - `sentence` - Must occur within the same sentence
  - `paragraph` - Must occur within the same paragraph
  - `page` - must occur within same HTML document (default)
- `<Suffixes>` Suffix processing for the search. Possible values:

  - `0` - Exact Match only
  - `1` - Plurals and Possessives
  - `2` - All Word Forms
  - `3` - Custom

- `<Thesaurus>` Set to `1` if the Thesaurus was used for synonyms
- `<Order>` Ordering of the search. Possible values:
  - `r` - relevance
  - `dd` - newest first
  - `da` - oldest first
- `<RankOrder>` Favors results with query terms in the same order as the query
- `<RankProximity>` Favors results with query terms close together
- `<RankDatabaseFrequency>` Favors results with query terms more rare across the entire profile
- `<RankDocumentFrequency>` Favors results with query terms repeated more often
- `<RankPosition>` Favors results with query terms earlier in the document
- `<RankDepth>` Favors results fewer links away from the starting point
- `<RankDateBiasWeight>` Date biasing: weight to favor newer results. Present if non-zero.
- `<RankDateBiasHalfLife>` Decay rate of `<RankDateBiasWeight>`: age (in seconds) at which only half of it applies. Present if set by search user.
- `<RankDateBiasAnchor>` Date of theoretical ”newest” (best) possible (full weight) result for date biasing. Present if set by search user.
- `<RankDateBiasField>` Field to use to compute age of documents for date biasing. Present if set by search user.
- `<mode>` Set to `admin` if this is a Test Search
- `<opts>` Internal use only
- `<authUser>` User that was authenticated via the Proxy Module
- `<metasearchTarget>` Indicates what backend metasearch targets are available, one element for each target. Currently selected targets will have a `selected="selected"` attribute
- `<AdminUrl>` URL to the admin interface
- `<MakeLiveUrl>` URL to make this Look and Feel live
- `<RssUrl>` URL to RSS version of this search
- `<OpensearchUrl>` URL to the OpenSearch version of this search

- <OpensearchTitle> Suggested title for this OpenSearch
- <QueryAutocomplete> Set to Y if Query Autocomplete is enabled
- <LogoutUrl> URL for a 'Logout' link
- <Category> Categories available for search
  - <CatVisible> Set to Y if the category should be selectable in the list of categories
  - <CatSel> Set to Y if this category is currently selected
  - <CatVal> Value to submit to search for this category
  - <CatName> Display name for this category
- <TopBestBets> List of "Best Bets" links
  - <BBTitle> Title for this section of Best Bets
  - <BestBet> Individual Best Bet records
    - <BBResultNum> Ordered number for this Best Bet
    - <BBPriority> Priority for this Best Bet, as assigned in the admin interface
    - <BBLink> URL for this Best Bet
    - <BBLinkDisplay> URL that displays for this Best Bet. Long Urls are intelligently truncated for display
    - <BBResult> URL for this individual Best Bet, as assigned in the admin interface
    - <BBDescription> Description for this individual Best Bet, as assigned in the admin interface
    - <BBGroupname> Name of the Best Bet group this Best Bet belongs to
    - <BBGroupid> id of the Best Bet group this Best Bet belongs to
    - <BBKeywords> Keywords that trigger this Best Bet record to display. This is all keywords for this individual record, not just the one that triggered this activation
- <ProfileInfo> Encloses some profile summary info
  - <Profile> Profile to which this ProfileInfo refers to
  - <Feature> Notes whether a feature is enabled: feature name is name attribute (e.g. proximity), enabled if isEnabled attribute is Y
  - <ResultDecl> Declarations of User Fields that will be in Result elements, each has a name and type attribute
  - <ExitIsEarly> Set to Y if search aborted
  - <ExitReason> Set to ok if search finished normally, otherwise token indicating reason (see ExitReason table below)
  - <RedirectUrl> Only used when results **Authorization Method** is set to Forward login cookies or CAS. If present, specifies a (%REFERER%-modified) version of **Login URL** (the search setting, not XML element). Its value is an external (not Webinator) URL to redirect the user to, which will prompt the user to log in and obtain the authentication cookies or parameters needed for a Results Authorization search.

  - <LoginUrl> Only used when results **Authorization Method** is set to Basic/NTLM/file - prompt via form. If present, specifies a local (Webinator) <form action> URL which will prompt for (and accept) the rauser/rapass variables, which contain user credentials needed for a Results Authorization search.

- <Summary> Encloses search results summary, only present if a search was actually performed
  - <Profile> Profile that this Summary element applies to
  - <Start> First result item to list
  - <End> Last result item to list
  - <TotalNum> Total number of result items found, *before* Results Authorization
  - <TotalIsEstimate> Set to Y if TotalNum is an estimate
  - <TotalIsShort> Set to Y if TotalNum is known to be short (e.g. early exit)
  - <UserResultsNum> Total number of result items found, *after* Results Authorization
  - <UserResultsIsEstimate> Set to Y if UserResultsNum is an estimate
  - <UserResultsIsShort> Set to Y if UserResultsNum is known to be short (e.g. early exit)
  - <ResultsAuthorization> Set to Y if Results Authorization was used
  - <Total> Readable text for total number of results, *after* Results Authorization
  - <GroupBySite> Set to Y if Results per Site was used with this query.
  - <CurOrder> Text that describes the order by which results are listed
  - <OrderLink> URL that provides an alternative sorting order results list
  - <OrderType> Text that describes OrderLink
  - <NewSkip> (Metasearch only) Skip value to use for any further request. Only needed with the SOAP API
  - <PreviousLink> URL to the previous page of results
  - <FirstPage> Set to 1 if this is the first page of results
  - <Pages> Contains data on pages of results
    * <PageLink> URL to a certain page of results
    * <PageNumber> Page number a page of results
  - <NextLink> URL to the next page of results
  - <LastPage> Set to 1 if this is the last page of results
  - <Credit> Text to introduce the credit image
  - <CreditImage> URL of the credit image

- <Result> Contains data about a given result
  - <Profile> Profile for this Result
  - <BackendProfile> Profiles used by metasearch backends
  - <Num> Number of this result item

- <Skip> Internal use: raw skip(s) for result. Valid for Meta Search back-ends
- <Id> Identifier for this result
- <ResultTitle> Title of this result
- <Url> URL of this result
- <ClickUrl> URL for this result item, as should be clicked by the user. Use Url if not present. Only sent if Query Logging is enabled, in which case it contains redirect for logging the click-through
- <UrlPDFHi> URL to highlight this PDF in Acrobat Reader, only used with Legacy highlighting
- <UrlDisplay> Displayed URL for this result
- <UrlWalk> URL used during the walk, if different from <Url>. Only used when a custom Result URL Source is set.
- <UrlCached> URL to retrieve the cached version of this result
- <RawRank> Raw relevance rank value for this result (0-1000)
- <ScaledRank> Raw rank scaled up for a more-like-this search (0-1000)
- <PercentRank> ScaledRank as a percentage (0-100)
- <DocSize> Size (bytes) of this result
- <MimeType> MimeType for this result
- <MimeTypeIcon> Icon file to use for this MimeType
- <Depth> Number of links walked from Base URL(s) to this URL
- <UrlSimilar> URL to search for pages similar to this result
- <UrlInfo> URL for context of answers within a matching document
- <UrlParents> URL of pages that link to this search result
- <Modified> Date and time this result was last modified
- <Visited> Date and time this result was walked
- <Abstract> Brief text surrounding the matched word or phrase
- <Charset> Character set of the formatted text of the page (typically Storage Charset unless conversion failure)
- <SiteName> Name of the site for this result item
- <UrlMoreResultsFromSite> URL for more results from this site
- In addition, any Additional Fields that have been selected for Output will be sent as child elements of Result, one per field. Each element is named after the field, with a u: XML namespace prefix since they are custom fields. The value of the field will be the content of the element.

  For example, an Integer field Quantity and a GMLPoint field Location may be given as:

```
<u:Quantity>57</u:Quantity>
<u:Location>47.4500 -122.3000</u:Location>
```

- <RightBestBets> List of right "Best Bets" links, see TopBestBets
- <Spelling> Spelling suggestions
  - <SuggestWord> An individual spelling suggestion
    * <SpellPhrase> Label for the suggestions
    * <SpellLink> URL to search for the suggestion
    * <SpellWord> Suggestion content
    * <SpellCount> Number of results for this suggestion
- <exportVar> Additional exported variables
- <QueryMessage> Messages to show to the user
- <Message> Additional diagnostic messages

  Attributes:

  - @type - Set to user for messages meant for end users, admin for Webinator administrator diagnostics
  - @code - Code for this message
  - @script - Script of this message
  - @line - Line number this message occurred

Table 5.2: XML <ExitReason> Tokens

Token	Description
ok	Normal exit
ResAuth-ExternalLoginRequired	Need Login Cookies: redirect to <RedirectUrl>
ResAuth-CredentialsRequired	Need user/pass: send rauser/rapass to <LoginUrl>
ResAuth-LoginIncorrect	User/pass incorrect; re-send to <LoginUrl>
ResAuth-SuccessLimit	Successful Auth Result Limit reached
ResAuth-Timeout	Results Authorization timeout
ResAuth-MaxDocsCheck	Max Docs to Auth-Check exceeded
NoProfileSpecified	No profile specified
InvalidProfileName	Invalid profile name (e.g. illegal characters)
NoSuchProfile	No such profile
Timeout	**Search Timeout** exceeded

Match Info output is similar to search results, except it contains a ContextResult element instead of Result elements. ContextResult contains:

<ContextResult> Container for the "Match Info" for this result

- <Url> URL of this result
- <ClickUrl> URL for this result item, as should be clicked by the user. Use Url if not present. Only sent if Query Logging is enabled, in which case it contains redirect for logging the click-through
- <UrlDisplay> Displayed URL for this result
- <Depth> Number of links walked from Base URL(s) to this URL, with a full text label
- <Size> Size (bytes) of this result
- <MimeType> MimeType for this result
- <MimeTypeIcon> Icon file to use for this MimeType
- <Modified> Date and time this result was last modified
- <Visited> Date and time this result was walked
- <RecordCategory> Categories that would match this result
- <Title> Title of this result
- <Description> Description of the result
- <Keywords> Keywords of the result
- <Meta> Extracted metadata of the result
- <Body> Body text the result
- In addition, any Additional Fields that have been selected for Output will be sent as child elements of Result, one per field. Each element is named after the field, with a u: XML namespace prefix since they are custom fields. The value of the field will be the content of the element.

  For example, an Integer field Quantity and a GMLPoint field Location may be given as:

```
<u:Quantity>57</u:Quantity>
<u:Location>47.4500 -122.3000</u:Location>
```

Invoking Query Autocomplete

Query Autocomplete can be used in your own custom front end using JavaScript similar to that used by the normal search interface. If you want to invoke it arbitrarily, you can request URLs of the form:

http://*HOSTNAME*/texis/webinator/search/autocomplete.json?pr=*profile*&term=*term*

Where *HOSTNAME* is the IP/hostname of your Webinator, *profile* is the profile to search, and *term* is the user's partially typed term. The URL path for your installation may be different from /texis/webinator/search, use whatever normally appears in your search URLs.

Autocomplete returns a JSON array in the OpenSearch format (http://www.opensearch.org/Specifications/OpenSearch/Extensions/Suggestions). Getting completions for `term=sea` would return something like:

```
["sea",["seattle","sears","search"]]
```

Autocomplete also supports JSON-P, so adding `&callback=updateList` to the URL would return:

```
updateList({term: "sea", completions: ["seattle","sears","search"]})
```

Alternatively, you can request `autocomplete.xml` instead of `.json` to get an XML document back:

```
<Completions>
  <Term>sea</Term>
  <Completion score="16">seattle</Completion>
  <Completion score="5">sears</Completion>
  <Completion score="1">search</Completion>
</Completions>
```

5.13.4 Invoke the search SOAP API

Instead of making a HTTP request and parsing the XML response, it's possible for pages to invoke the search SOAP API. This allows interactions with the Webinator to appear as local function calls, automatically handling all the details of HTTP requests and XML parsing.

See the **SOAP API** (p. 177) for details on setting up and using the SOAP API.

Sample ASP Code

The `samples` directory of your Webinator installation contains `search.asp`, which demonstrates sending an http GET command to Webinator and receiving XML search results from it.

5.14 Search Result RSS Feeds

Search result RSS feeds can help you monitor a certain search query, and let you know when new results appear for the query.

All search result pages have an RSS link embedded in them. Recent versions of modern browsers, such as Internet Explorer and Firefox, have built-in features that notify you when an RSS feed you're subscribed to changes.

- IE 7 and 8 - `http://www.microsoft.com/windows/IE/ie7/tour/rss/`

- Firefox - `http://kb.mozillazine.org/Live_Bookmarks_-_Firefox`
- Opera - `http://www.opera.com/mail/rss/`

5.15 OpenSearch Support

The search interface also has an embedded Open Search description. This means that modern browsers can use the Quick Search box (to the right of the address bar) to perform searches on Webinator.

- Bring up the search interface for the profile of your choice
- Hit the ”down” arrow next to the Quick Search box
- Choose “Add Search Provider...” to add Webinator to the list of available searches.

Internet Explorer users can find more detailed instructions at
`http://msdn.microsoft.com/en-us/library/cc848862.aspx`

Adding `strictSpec=Y` to the Open Search Description URL will cause the Webinator to truncate various fields as required by the specification. In practice, this isn't necessary as browsers handle longer names.

5.16 Using Best Bets

Webinator allows you to create links that will appear either at the top or to the right of the search results (or anywhere else, if using an XSL stylesheet) when specific keywords are searched for. They can be used for suggested links, or to promote specific URLs so they stand out from the main results. The Best Bet links are arranged into groups, which allow you to enable or disable a group of results easily.

5.16.1 Quick Creation

The easiest way to create Best Bets is to directly add keywords to URLs. This skips the group and display settings, which can be customized later (and are detailed below).

From the “List/Edit URLs” page, enter the desired URL and click on the URL to get the details on that URL. There is a form on the page that allows keywords to be added to that URL. You can define a priority, title, description, and keywords for the URL (as detailed in the list below, under **Fully Customized**).

The group will be listed as `(Create New)`. This will create a default group and automatically set it to display, instantly using the Best Bet you just created. The created group (`default`) can then be used to create any number of other keyword-URL associations.

You can go to the “Search Settings” page to customize how the Best Bets are displayed, as detailed below.

5.16.2 Fully Customized

Best Bets can also be created fully customized. The first step is to define a group. This is done from the "Bestbet Groups" page under "Tools". You can name the group, and decide which information will be displayed about the group.

After creating a group, you can use, the "Manage BestBets" link to add Best Bets to the group. You can also browse "List/Edit URLs" page enter the URL you want, and click on the URL to get the details on that URL. The fields on the form are:

- **Url** - The URL to link to with this Best Bet. Only used when adding Best Bets directly to the group, rather than going through "List/Edit URLs'.
- **Priority** *(optional)* - An integer priority for this Best Bet. If multiple Best Bets match a given user query, they are shown in descending numerical **Priority** order. If there is only one Best Bet set per URL, or the order does not matter, a **Priority** need not be set.
- **Title**- The title that will be displayed for the Best Bet on the search results page.
- **Keywords**- A space-separated list of keywords that will be searched against to trigger this Best Bet. A Best Bet is displayed when the user query matches the **Keywords**, just as if they were document text.
- **Group**- Which Best Bet Group this Best Bet will be created in. A Best Bet will only have a chance to match if its group is set to display as either **Top Best Bets** or **Right Best Bets**on the Search Settings page.

  If no groups currently exist, `(create new)` will be displayed, and a group will be created for you if you enter keywords and a title for this Best Bet.

  Only used when adding BestBets through **List/Edit URLs**, rather than adding to a Best Bet group directly.
- **Description** *(optional)* - The description to display for this Best Bet. The Best Bet Group for this Best Bet might be set to not display the description, so it's optional.

The title and description can contain HTML code. Be careful that it does not disrupt the rest of the page layout. You can create multiple entries for the same URL. Each time you save a new set of blank boxes will be shown.

Once the Best Bets are created you can go to the "Search Settings" page to set up how they are displayed. For the top and right placements you can define which group is shown there, what title if any to display above the links, and the color, size and style of the boxes around the Best Bets.

As with any of the Search Settings these will apply to the "Test Search" first, and then when you apply the settings be copied to the "Live Search", allowing you to test the settings and make sure they are appropriate before going live.

5.17 Using Access Control

The concepts and actions of access control in Webinator are discussed in detail in the Access Control section, p. 134. The following are some general tips on how to setup and maintain access control rights.

Access Control is supported in the full Texis product, but not Webinator-only.

5.17.1 Initial Lockdown

Since the default mode for Access Control when created is to allow all rights to all users for back-compatibility, it is recommended that perms be "locked down" first, and only granted as needed. The `webinator` user, having the irrevocable ability to reset ACLs, should remain a "superuser" with all access, and other accounts turned into lesser-permission users. Lockdown should happen in this order:

1. Allow superuser: The `webinator` user should have an `Allow` entry for all rights to the top-level `Global` object[1].
2. Deny everyone: The group `Everyone` should have a `Deny` entry for all rights to the top-level `Global` object.

With these perms, users other than `webinator` – including new users and profiles created in the future – will not be able to see or modify administrative settings. They can be granted perms as needed later, for example, the Read right could be removed from the `Global` deny ACE so that they can read but not modify any admin action/setting.

5.17.2 Example: User with Complete Control on One Profile

To configure a user that has complete access to just one specific profile (but no other profiles, nor the rest of administration such as creating accounts etc.), set up the lockdown settings above, then:

1. Create a Profile ACE on the specific profile, for that user, read and write access, and type `Allow`.
2. Create a Setting ACE for `All Settings`, for that user, read, write and delete access, type `Allow`.

The user will now be able to modify any setting on that profile, as well as start/stop walks on it, but will not be able to edit other profiles.

5.17.3 Example: User with Look and Feel Control on All Profiles

To configure a user that has the ability to change the Top and Bottom HTML on *any* profile, but cannot edit walk settings, nor start nor stop a walk, etc., set up the lockdown settings above, then:

[1] In version 5.3.0 and earlier, the webinator user should instead be explicitly granted all rights to each of the second-level objects (All Users, All Groups, All Profiles, All Settings, and Maintenance).

1. Create a Profile ACE on `All Profiles`, for that user, read and write access, and type `Allow`.
2. Create a Setting ACE for `Top HTML`, for that user, read, write and delete access, type `Allow`.
3. Create a Setting ACE for `Bottom HTML`, for that user, read, write and delete access, type `Allow`.

The user will now be able to change the top and bottom HTML for any profile.

5.18 Replication

5.18.1 Replication Overview

In replication, a server profile sends walk data to another server profile. The two profiles can be on different machines or they can be on the same one. If the profiles are on different machines, the sending and receiving profiles can have the same or different names. If the profiles are on the same machine, use different profile names.

Replication is supported in the full Texis product, but not Webinator-only.

Here is an example that illustrates the replication process. In this example, the `Sender` profile has been set up as the sender profile and `Receiver` is the receiver profile. After `Sender` performs a walk, it sends the walk data to `Receiver`. The `Receiver` profile accepts the data as-is, without regard to its own profile settings. Only the profile that performed the walk may send the walk data, so in this example `Receiver` cannot replicate (the data it received from `Sender`) to another profile.

To avoid undesired overwriting of replication walk data, you should not allow the receiver profile to perform walks.

Before the receiver will accept replication data, the sender(s) need to be granted permission to send the data. This permission is managed in the **Cluster Members** list.

A good use of replication is to set up multiple machines to replicate to a single receiving profile. For example, machines A, B, and C each have a different profile, and they each replicate their walk data to a profile on machine D, which is the receiver. Another use of replication is to send walk data from multiple profiles on a machine to a single receiver profile that is on the same machine. This provides a means of combining walk data into a single profile. Another use of replication is to replicate data from one sender to multiple receivers. This way multiple machines hold the same walk data.

5.18.2 Procedure - Replicating One Profile

The procedure in this section is an example of setting up replication on a single machine for a single profile. See the next section (p. 167) for an example of backing up all profiles on one machine to another machine.

Set up the Sender Profile

- Choose an existing, walkable profile to be the sender. Or go to the `Profiles` menu item and create one, filling in all fields for a normal walk. We'll assume this profile is called `Sender`.

- Go to the `All Walk Settings` menu item for the `Sender` profile.
- Scroll down to `Replication Settings`.
- Enter the information for the receiver. In this example, `Host IP or Name` is `localhost` because we'll be sending data to the same machine, and `Profile Name` is `Receiver`. The page now includes the location of the receiver profile.
- Click `Update and Go` button.
- After a moment, the `Walk Status` page opens. Notice that there are N items in the replication queue. The number N is similar to the number of pages that were walked. The items remain in the queue, because they cannot be sent until the receiver profile is created (below). Normally, when a receiver profile is present, the contents of the queue are automatically sent to the receiver.

Create the Receiver Profile

- Create a new profile called `Receiver` via the `Profiles` menu item. (This matches the receiving profile name we entered on the `Sender` profile.)
- At main menu click `System`, then under `System Settings` heading, click `System Wide Settings`.
- At the **Cluster Members** field, enter the IP address for each server that will send walk data to this machine. Use a separate line for each entry. In this example, there is one sending IP address, and it is 127.0.0.1 (use IP numbers, not the word `localhost`). To enable an entire subnet to send data, use an IP prefix and wildcard, e.g. `10.10.*`.
- Click `Update` button.
- At main menu, click `Profiles`.
- When `Profiles` page opens, click `Sender`. A `Walk Settings` page opens for the `Sender` profile.
- Click `Walk Status` button. The `Walk Status` page for the `Sender` profile opens.
- There are still N items in the replication queue.
- Click the `replication queue` link.
- The items in the replication queue are sent to the `Receiver` profile. On the `Walk Status` page, there are now 0 items in the replication queue, which indicates the items were sent.
- On main menu, click `Profiles`, click `Receiver`, click `Walk Status` and observe that there is a list of pages recently walked. These pages were not walked by `Receiver`, instead they were obtained from `Sender`, which performed the walk.

5.18.3 Procedure - Separate Hot Backup Machine

The following procedure uses System Replication to back up System-Wide settings, all profiles' settings, and all profiles' walk data from one machine to another. This could be used to set up a "hot backup" machine that automatically receives settings and data changes from a live machine. Once configured, during normal operations the backup machine will thus neither be further manually configured (e.g. profile settings changes), nor will it walk profiles, as it now automatically receives both from the main machine. If the main machine goes down, the backup can then be instantly swapped in as the live search, without waiting for restoration from backup or rewalks.

Configure the Backup Machine

On the backup machine:

- Under `System` → `System Setup` → `System Wide Settings`, add both the main and backup machines' IP addresses to **Cluster Members** (one per line). Adding the main machine IP will permit the backup machine to cooperate with the main machine (e.g. when receiving settings/data from it). Adding the backup IP will avoid needing to add it if the roles of the main and backup machines ever switch (e.g. after a disaster and the main machine is rebuilt and becomes the backup).
- On the same page, set **Disable Starting All Walks** to `Y`. This will prevent redundant (and possibly conflicting) walks on the backup, which are unnecessary because it will be receiving walk data from the main machine.
- On the same page, under **System Replication Settings**, set **Allow Receiving** to `Y`. This will allow the backup machine to receive replication data (i.e. settings and walk data) from the main machine.
- Hit `Update` to apply the above changes.

Configure the Main Machine

Next, on the main machine:

- Under `System` → `System Setup` → `System Wide Settings`, add both the main and backup machines' IP addresses to **Cluster Members** (one per line). While not strictly necessary immediately, this keeps the setting consistent with the backup machine, in case the two switch roles.
- On the same page, under **System Replication Settings / Targets**, add a target, and enter the backup machine's IP address. This will begin to send settings changes and newly walked pages to the backup machine.
- Hit `Update` to apply the above changes.

Synchronize Pre-existing Profiles

The hot backup is now configured, and will receive settings changes and newly walked pages going forward. No walks should be performed on the backup machine – indeed, we disabled them – nor should settings be further modified on it, as it is receiving both from the main machine automatically.

Since only *changes* to settings will be propagated by replication, any pre-existing, non-replication System-Wide Settings (e.g. HTTPS settings) on the main machine should be copied – just one time, now – to the backup machine to ensure it is in sync.

More importantly, if there are pre-existing profiles on the main machine, they must be propagated now. Otherwise future changes to those profiles will not propagate (and may cause replication to stall). This can be accomplished with the following steps:

- Go to `System` → `System Replication` → `System Replication Target Status` on the main machine: its current profiles will be listed, as well as whether they exist on the backup machine.
- If any profiles are not shown as "ok", they do not exist on the backup machine: they should be copied now, so that future changes and walks propagate. Simply hit `Create Missing Profiles` at the bottom of the page, and all these profiles' settings will be queued for replication to the backup.
- Re-visit the `System Replication Target Status` page in a few minutes to verify this: all profiles should eventually be "ok" under the backup machine column.
- The (previously missing) profiles' data should be copied too. This step can be skipped if those profiles will all be doing `New` walks in the near future on the main machine, as the data will be copied then. However, if `Refresh` walks are being used, or walks are rare, or simply to ensure the data is backed up now, the data can be propagated manually:
  - For each (previously missing) profile, choose that profile from `Profiles`.
  - Go to `Tools` → `Replication Tools`, choose the backup machine under **Send Profile Data**, and hit `Send Data`.
  - Wait for that profile's data to be sent – large walks can take a while – before proceeding to the next one: the status can be seen under the `View Replication Status` link; wait for it to become empty.
  - Repeat for other profiles.

This `System Replication Target Status` check can be occasionally performed in the future to ensure all profiles exist on the backup. However, it should not be needed past this initial setup stage, as *new* profiles (created after system replication is active) will be created automatically on the system replication target(s) configured earlier.

Making Backup Live on Main Failure

If in the future the main machine fails and the backup needs to become primary, follow these steps:

- Ensure that the main machine stops replicating, if it is still accessible, to avoid further confusion. Go to **System Replication Settings / Targets** on the main machine and remove all target(s), then hit `Update` at the bottom to apply this. If the main machine is inaccessible, simply make sure it stays down.
- Make the backup machine live for searches as appropriate (e.g. switch your organization's proxy or web site to refer to it, change its IP/hostname, etc.). We now refer to this as the new main machine.
- Turn off receiving replication on the new main machine: under `System` → `System Setup` → `System Wide Settings`, set **Allow Receiving** to `N`.
- Replace/restore the new backup (old main) machine, and configure as a backup per above.

5.18.4 Using Circular Replication

It's possible to have two Webinator installations replicating to each other. This can be useful behind a load balancer to automatically detect when one installation is down and start sending to the other one.

Content received through replication is not passed along to the receiver's replication targets, so they won't create an "echo chamber" sending the same request back and forth.

Setup

The setup is the same as `Separate Hot Backup Machine` above, but as a final step, you also configure the receiver as a sender, pointing at the original machine.

Notes and Limitations

If running walks, each profile should only run on one machine or the other at once. If you have a very large profile, it's recommended to split it into two smaller profiles, and then index one profile on one machine and one profile on the other, letting them replicate to each other.

Using circular replication behind a load balancer can make troubleshooting issues more difficult versus a standard "sender and receiver", as it's often unclear which one sent or received an individual piece of data.

5.18.5 Dataload API

The replication system can also be used to load data directly onto Webinator from an outside source, instead of "pulling" it from a URL or its links. This can be used for data that is not permanently stored at its URL (e.g. generated data), and therefore cannot be fetched for indexing; it can instead be pushed to Webinator for indexing.

The Dataload API is supported in the full Texis product, but not Webinator-only.

Before loading data onto Webinator, it must be configured to accept data from the IP address(es) that will be sending to it. This procedure is the same as for replication; see the **Cluster Members** setting, p. 166.

Submission Format

Data is submitted to Webinator with an HTTP POST request sent to a similar URL as the admin interface (e.g. `http://.../dowalk`), but with `/recvdata.xml` appended. E.g.:

```
http://www.example.com/texis/webinator/dowalk/recvdata.xml
```

The following POST variables must be set in the request. Be sure to URL-encode the values:

- `profile`
  Set to the name of the receiving profile.

- `data`
  Set to an XML document containing the data, and what to do with it (insert/delete/etc.). See below for details.

Specifying all fields manually

Below is an example `data` document where all fields are specified. Be sure to HTML-encode values.

```
<?xml version="1.0" encoding="UTF-8"?>
<ThunderstoneReplication
      xmlns:dt="urn:schemas-microsoft-com:datatypes"
>
  <Item>
    <Type>I</Type>
    <Size>150369</Size>
    <Visited>2005-10-25 15:25:18</Visited>
    <Dlsecs>0</Dlsecs>
    <Depth>0</Depth>
    <Url>http://www.example.com/dir/page.html</Url>
    <Title>Sprocket Specifications</Title>
    <Body>...</Body>
    <Keywords>sprockets, gears, hubs</Keywords>
    <Description>Sprocket details</Description>
    <Meta></Meta>
    <Category>Mechanical</Category>
    <Modified>2005-10-25 11:21:07</Modified>
    <NextCheck>2005-10-25 16:25:18</NextCheck>
    <Views>0</Views>
    <Clicks>0</Clicks>
    <CTR>0.000000</CTR>
    <Pop>0</Pop>
    <MimeType>text/html</MimeType>
    <Charset>UTF-8</Charset>
    <Refs dt:dt="bin.base64">...</Refs>
    <Errors dt:dt="bin.base64">...</Errors>
    <RawData dt:dt="bin.base64"></RawData>
  </Item>
</ThunderstoneReplication>
```

Any element whose text data might not be XML-safe (e.g. binary chars in the `<Body>`) should be base64-encoded, and the attribute `dt:dt="bin.base64"` set in the tag. E.g. the `<Refs>` and `<Errors>` elements' text data are always base64-encoded. Note that the XML namespace prefix `dt` should also then be set to `urn:schemas-microsoft-com:datatypes` in the root `<ThunderstoneReplication>` element.

The elements are:

- `<Type>` The action to take with this data. Text value may be one of:
  - `I` - Insert the data (overwrite all previous data for URL, if any)
  - `D` - Delete the URL
  - `DP` - Delete the URL as a pattern (e.g. `http://www.example.com/dir/*`)

  - `U` - Update the URL, leaving unspecified fields unchanged
  - `UI` - Update search indexes (call after a batch of inserts/deletes)

- `<Size>` The integer size of the original document.
- `<Visited>` When the document was fetched, in `YYYY-MM-DD HH:MM:SS` format.
- `<Dlsecs>` Number of seconds taken to download the document.
- `<Depth>` Depth of URL from a Base URL, e.g. 0 is a Base URL, 1 is one click away, etc.
- `<Url>` The URL of the document.
- `<Title>` The title of the document.
- `<Body>` The formatted body of the document.
- `<Keywords>` Any keywords for the document.
- `<Description>` The description of the document.
- `<Meta>` Any meta data for the document.
- `<Category>` The category the document is in, if any. Must be a category name from the profile's Categories.
- `<Modified>` The `Last-Modified` date of the document in `YYYY-MM-DD HH:MM:SS` format.
- `<NextCheck>` When the document should be refreshed, in `YYYY-MM-DD HH:MM:SS` format.
- `<Views>` Number of views of the document: how many times it has been shown in search results.
- `<Clicks>` Number of clicks of the document: how many times it has been clicked on in search results.
- `<CTR>` Click-through-ratio: floating-point number ratio of clicks to views.
- `<Pop>` Document popularity: number of references (links) to it.
- `<MimeType>` The MIME type of the content served at the URL, or provided in RawData.
- `<Charset>` Character set of `<Body>` data. Should correspond with `Storage Charset` profile setting (p. 69). If a charset other than the `Storage Charset` is used, it should be a standard IANA charset that Webinator can convert to the `Storage Charset`.
- `<Refs>` Optional element with references (child links) of the document.
- `<Errors>` Optional element with errors of the document.

Uploading a binary file

If you have a binary file, such as a PDF or an Office document, you can send it with the dataload API and let the Webinator extract the text from it.

```
<?xml version="1.0" encoding="UTF-8"?>
<ThunderstoneReplication
      xmlns:dt="urn:schemas-microsoft-com:datatypes"
>
    <Item>
        <Type>I</Type>
        <Url>http://www.example.com/dataload.pdf</Url>
        <RawData dt:dt="bin.base64">0M8R4KGxGu....</RawData>
    </Item>
</ThunderstoneReplication>
```

The elements are:

- `<Type>` The action to take with this data. Text value may be one of:
  - `I` Insert the data (overwrite previous data for URL if any)
- `<Url>` The URL of the document.
- `<RawData>` element with the base64 encoding of raw document. It must include the `dt:dt="bin.base64"` attribute.

Combining the two: binary files with custom fields

It is possible to specify both a `<RawData>` document, *and* fields such as `<Title>`, `<Description>`, etc. The binary document will be processed, and any other fields provided will override the values that came from the document.

This can be useful in situations where you have a Content Management System (CMS) that contains metadata about a document that doesn't actually *occur* anywhere in the document. You can do a custom dataload that pushes in the document, and the custom Title/Description/etc.

Additional Fields

Each profile-specific Additional Field is optionally sent in a single element named after the field, with the XML namespace prefix `u`. The value of the field is the content of the XML element. Note that the `u` XML namespace prefix should be declared in the root `<ThunderstoneReplication>` node, as shown earlier.

For example, an `Integer` field `Quantity` and a `Text` field `State` may be given as:

```
<u:Quantity>57</u:Quantity>
<u:State>NY</u:State>
```

Other Details

The optional <Refs> element lists the links (references) from the given document, for parent-child linking. Its text value is a base64-encoded XML document with the following format when decoded:

```
<results xmlns:dt="urn:schemas-microsoft-com:datatypes">
  <result>
    <Url>http://www.example.com/dir/page.html</Url>
    <Ref>http://www.example.com/dir/otherpage.html/</Ref>
  </result>
  ...
</results>
```

Each <Url> should be the same as the <Url> in the above <Item> block. The <Ref> is a single link from the page. Only one <Ref> may be listed per <result>; additional links should be sent with additional <result> elements.

The optional <Errors> element contains any errors to be logged for the document. Note that this may be empty or not present if no errors are to be logged. Its text value is a base64-encoded XML document with the following format when decoded:

```
<results xmlns:dt="urn:schemas-microsoft-com:datatypes">
  <result>
    <Url>http://www.example.com/dir/page.html</Url>
    <Reason>Document not found: 404 (Not Found)</Reason>
  </result>
  ...
</results>
```

As with the <Refs> element, the <Url> must correspond with the original <Item> <Url>, and multiple errors must be listed in separate <result> elements.

Reply Format

The response to a Dataload request is an XML document:

```
<ThunderstoneReplicationResult>
  <ItemResult>
    <rid>000000000</rid>
    <Type>D</Type>
    <DP>1</DP>
    <Status>OK</Status>
    <Info>Not found</Info>
  </ItemResult>
  <Rows>1</Rows>
  <Version>Version 5.01.1234567890 20051010 (...)
           2005-10-10 12:34:56</Version>
</ThunderstoneReplicationResult>
```

The elements are:

- `<rid>` The replication id. Ignored.
- `<Type>` The action type specified in the request.
- `<DP>` The number of URLs deleted by a `<Type>DP</Type>` action. Element is not present for other `<Type>`.
- `<Status>` Result code:
  - `OK` Success
  - `FAIL_UNKNOWNTYPE` The `<Type>` was not recognized
  - `NODATA` No parseable data in request
  - `Not Allowed` Sender is not in **Cluster Members**
  - `No Profile` No `profile` set in request POST
  - `FAIL` Failed, unknown reason
- `<Info>` Optional additional message; e.g. `Not found` if a non-existent URL is deleted
- `<Rows>` How many request `<Item>`s were processed.
- `<Version>` Version and release date of the software.

Once data has been successfully loaded onto Webinator, if the profile has any receiver profiles defined under `Replication Settings`, the data will also be queued for replication to those receivers.

Dataload SOAP API

There is a SOAP API available for dataload, allowing you to use a SOAP library to communicate with the Webinator. For an overview of SOAP, Please see the **SOAP API** (p. 177).

The WSDLs for the dataload API can be found on the `Profile Tools` page. Providing these WSDLs to whatever tool your language uses, such as Visual Studio's `wsdl.exe` program, should generate the necessary wrapper class.

The parameters are defined within the WSDL itself, and are generally the same as mentioned above in the **Submission Format** and **Reply Format**s, with a few exceptions:

- The entire transactions are wrapped by SOAP envelops and the top-level elements are called `dataload` and `dataloadResponse` instead of `ThunderstoneReplicationResult`, respectively.
- The `dataload` element contains a `profile` element in addition to all the `Item`s.

C# Example Project

A C# example project is available that demonstrates using both the search and dataload SOAP interfaces. You can find these projects in the `texis/samples` directory of your installation.

5.19 Additional Fields

5.19.1 Overview

The additional fields feature in Webinator allows you to define structured data that can be searched on, sorted by, and included in the results when using an XSL stylesheet. Typical uses might include having prices, dates or ratings associated with the documents.

Additional Fields are supported in the full Texis product, but not Webinator-only.

5.19.2 Populating

To populate Additional Fields they should first be defined in the **Additional Fields** section of **All Walk Settings**. You can specify a name, which is used as the name of the XML element when displaying the results, as well as when using the Dataload API.

Once the field has been defined it can be populated either via the Dataload API or through the **Data From Field** settings section. The fields are positionally numbered, and you can load Extra Field 1, 2 and/or 3 from the page that is read. If you are loading from a `<meta>` field you will typically want a **REX Search** of `.+` and the **From Meta Field** you are loading from.

5.19.3 Sorting

To sort results by an Additional Field, use the `order` form variable. To specify the first Additional Field set the value to `af1`, for the second `af2` and for the third `af3`. To reverse the sort order you add a `d` to the value, i.e. `af1d`, `af2d`, `af3d`.

5.19.4 Searching

See the **Additional Fields** section in **Issuing a Query Programmatically** (p. 152) for information on specifying additional field searches in the URL.

5.20 SOAP API

5.20.1 SOAP Overview

The Simple Object Access Protocol (SOAP) is a W3C (World Wide Web Consortium) recommendation that essentially allows for Remote Procedure Call (RPC) functionality over HTTP, via XML. (This is a simplification of the 120-page SOAP spec, but it suits our purposes). SOAP web services provides a systematic, defined way of communicating function requests and responses over a network transport.

SOAP interfaces are described by another W3C recommendation, WSDL documents - Web Services Definition Language. WSDL documents are the prototypes for SOAP functions. They define what parameters are expected to the functions, what formats are/aren't allowed, what will be returned, etc. Given the WSDL of a SOAP web service, programs can generate the client code that interacts with the services (as is demonstrated in the C# example project later).

Specifically for the Webinator, the SOAP interface provides, when using a language that has a SOAP API, a way to invoke a search and on the Webinator and insert data as if it were a local function call.

5.20.2 SOAP API vs. XML Output

The SOAP interface provides functionality very similar to the "XML output" search interface. So why use one as opposed to the other?

Use SOAP if the language you are writing has a SOAP interface available for you. Many languages and environments (including Visual Studio) provide SOAP tools, where you provide the WSDL to the webservice, and it will generate "wrapper" classes for you, allowing you to interact with Webinator as if it were simply a local function.

If whatever development environment you're using doesn't have a real SOAP interface, then use the XML API instead of the SOAP API. All the added information/rules of SOAP that make it easy for programs to exchange data will instead make it more cumbersome to use manually.

5.20.3 Getting the WSDL

The WSDLs can be found on the `Soap Tools` page for the profile. It links to the Dataload WSDLs and a search WSDLs, which lets you choose either a WSDL for this profile, or for all profiles (as explained below).

5.20.4 Global vs. per-profile WSDLs

When viewing search WSDLs, you have the option of requesting a WSDL specific to a single profile, or a global `All Profiles` WSDL, which can be used for any profile.

If you do not make use of Additional Fields, then there will be no difference between per-profile and global WSDLs.

Both per-profile and global WSDLs refer to the same search interface. The same SOAP response is generated for both WSDLs. The only difference is in how specific the WSDLs are - per-profile WSDLs specify which Additional Fields occur in the results, but the global WSDL must use `<xsd:any>` as a catch-all, as the Additional Fields may change from one profile to another.

Which you use is a trade-off that you must decide on.

- **per-profile WSDLs**
  - **Advantage** Additional Fields for the profile are "hard-coded" in the WSDL itself, so a SOAP client consuming the WSDL can make better use of the Additional Fields.

    For example, if your profile has Additional Fields called `price` and `location`, then a per-profile WSDL will specify that each result contains `<price>` and `<location>` elements. WSDL tools can do things like declare `response.price` and `response.location` variables.
  - **Disadvantage** Because the per-profile WSDL is specific to that profile's Additional Fields, a different WSDL must be used for every profile you want to interact with. If you're interacting with many different profiles (or it often changes), an global WSDL may be better suited.
- **Global WSDLs**
  - **Advantage** The `All Profiles` wsdl can be used for any profile. This is better if your application needs to query multiple profiles, or if you don't work with Additional Fields.
  - **Disadvantage** Additional Fields are represented in the `All Profiles` WSDL with `<xsd:any>`, which allows it to not declare which Additional Fields will occur in the XML (as it may change from one profile to another).

    This means that programs consuming the WSDL cannot know which Additional Fields will be returned, and will instead do things like offer an array of Additional Field XML elements that you must manually loop over to find the ones you want.

5.20.5 Configuring the SOAP Interface

The WSDL for Webinator is accessible in the following URL path from Webinator:

`/texis/ThunderstoneSearchService/describe.wsdl`

This link is also available from the `Tools` → `SOAP Tools` menu page in the admin interface.

Dataload SOAP API

The Dataload SOAP API takes the same parameters as the normal dataload API, please see the `Submission Format` (p. 170) and `Reply Format` (p. 175) sections, with a few exceptions:

- The entire transactions are wrapped by SOAP envelopes and the top-level elements are called `dataload` and `dataloadResponse` instead of `ThunderstoneReplicationResult`, respectively.
- The `dataload` element contains a `profile` element in addition to all the `Items`.

5.20.6 C# example project

A C# example project is available that demonstrates using the search SOAP interface. You can find `SoapSearchExample.zip` in the `texis/samples` directory of your installation.

5.20.7 SOAP Links for Languages

This section contains links for recommendations of SOAP implementations in other languages. Thunderstone makes no guarantees to the completeness or quality of these projects, we simply provide links for convenience.

- ASP.NET - the same C# API code can be compiled into a .NET assembly and used from an ASP.NET page. Please see the Windows Communication Foundation documentation for more details.
  - `http://msdn.microsoft.com/en-us/library/dd456779.aspx`
- Perl - SOAP::Lite for Perl is a collection of Perl modules which provides a simple and lightweight SOAP interface.
  - `http://www.soaplite.com/`
- Python The Web Services for Python Project provides libraries implementing the various protocols used when writing web services including SOAP, WSDL, and other related protocols.
  - `http://pywebsvcs.sourceforge.net/`
- Java - The Java API for XML Messaging (JAXM) provides a framework for sending and receiving SOAP messages.

  - `http://java.sun.com/webservices/jaxm/`

- C++ - Apache Axis C++ can be used as a client for SOAP servers.

  - `http://axis.apache.org/axis/cpp/clientuser-guide.html`

5.20.8 SOAP API search Reference

Three of the functions for searching (`search`, which performs a normal search, `moreLikeThis`, which finds similar pages, and `showParents`, which shows which pages link to a page) take similar parameters:

- `jump` - number of user-visible results to skip. 0 would return the first page of results, 10 the second page, etc. (assuming 10 results per page).
- `order` - specifies how the results should be ordered. Possible values are: `r` - sort by relevance (default) `dd` - newest pages first `da` - oldest pages first

RankKnobs structure

There's an optional "rankKnobs" parameter for many of the functions that can specify how things should be ranked (each function notes whether it accepts rankKnobs). All of these can be set from 0-1000, where the higher the value, the more heavily that aspect is weighed; 500 is the default. These parameters correspond directly to the "Ranking Factors" settings on the Advanced Search page.

RankKnobs has the following parameters:

- `order` - importance of the words being in the proper order
- `proximity` - importance that the words are close together
- `dbFreq` - importance of the frequency of a word in the database
- `docFreq` - importance of the frequency of a word within the document
- `leadBias` - importance of closeness to the start of the document
- `depthBias` - importance of "shallowness" (fewer links from a **Base URL**)
- `dateBiasWeight` - Date bias weight: Favors "newer" results (closer to `dateBiasAnchor` i.e. `now`). Additional parameters:
  - `dateBiasHalfLife` - Date bias decay rate: time for `dateBiasWeight` to be halved, in seconds
  - `dateBiasAnchor` - Date bias reference point: "best" date for maximum rank; can be `lastWalkFinished` for completion date of last successful walk, or Texis-parseable date
  - `dateBiasField` - Date bias field: date field to use for computing document age (default `Modified`)

search

This performs a normal search based on the query/queries provided.

Parameters:

- `profile` - Required. Specifies which profile you're working with.
- `query` - Required. The Metamorph query to search for.

The following parameters can be provided to further refine the search:

- `urlQuery` - Used for URL Prefix queries. Corresponds to the default search interface's `uq` query-string variable (p. 150).
- `depthQuery` - the maximum depth that would be allowed. Supplying a value of `3` would only return pages that are no more than 3 clicks away from a Base URL. Corresponds to the default interface's `dq` variable.
- `categoryName` - name of a category to limit results to. Corresponds to the default interface's `category` variable. Added in version 20.1.
- `categoryQuery` - numeric index for a category to limit results to. 1 is the first category, etc. Corresponds to the default interface's `cq` variable. Deprecated; use `categoryName` instead.
- `requireAllCategories` - Set to `Y` to require each result to match *all* categories specified, instead of any of them.
- `proximity` - specifies a required proximity for the words in the query. Corresponds to the default interface's `prox` variable. Possible values are:
  - `page` - words must occur on the same page (default)
  - `paragraph` - words must occur in the same paragraph
  - `sentence` - words must occur in the same sentence.
  - `line` - words must occur on the same line.
- `authUser` - Username to use for Results Authorization, when using the `Basic/NTLM/file - prompt via form` authorization method.
- `authPass` - Password to use for Results Authorization, when using the `Basic/NTLM/file - prompt via form` authorization method.

The `search` function may also use the rankKnobs structure (section 5.20.8, p. 180).

Additional Fields

The `search` function may also take a number of Additional Field parameters, as described in the `Searching Additional Fields` section (p. 62).

Response

The SOAP output of the function is described in the XML Elements in Search Results section (5.13.3, p. 153).

moreLikeThis

`moreLikeThis` returns results that are similar to a result already found.

Parameters:

- `profile` - Required. Specifies which profile you're working with.
- `id` - Required. the id of a single URL, returned from a previous function.

The `moreLikeThis` function may also use the `<rankKnobs>` structure, as described in section 5.20.8, p. 180.

The output of the function is described in the XML Elements in the Search Results section (5.13.3, p. 153).

matchInfo

`matchInfo` gives information about the result, such as description, keywords, and the body text.

Parameters:

- `profile` - Required. Specifies which profile you're working with.
- `id` - Required. the id of a single URL, returned from a previous function.
- `query` - Providing the query will allow the Webinator to highlight the query in the match info response.

The output of the function is described in the XML Elements in the Search Results section (5.13.3, p. 153).

showParents

`showParents` lists all the pages that link to a previous retrieved search results.

Parameters:

- `profile` - Required. Specifies which profile you're working with.
- `id` - Required. the id of a single URL, returned from a previous function.

The output of the function is described in the XML Elements in the Search Results section (5.13.3, p. 153).

getCompletions

`getCompletions` retrieves potential completions for a partially typed query. It can be used by a custom client to implement autocomplete as the user is typing.

Parameters:

- `profile` - Required. Specifies which profile you're working with.
- `term` - Required. The partially typed term to get completions for.

Output: Zero or more `Completion` elements are returned. Their text content is a potential completion, and each completion has attributes:

- `profile` - the profile this completion came from.
- `score` - the score for this completion. Completions are ranked according to score, which can be derived from occurrences in the content, queries, and search user clicks.

5.20.9 SOAP API dataload reference

The dataload SOAP API is very simple. There's only one function, which is for loading data into the Webinator.

dataload

Parameters:

The SOAP input for dataload is described in the *Submission Format* section of the Dataload documentation (5.18.5, p. 170).

Returns: The SOAP output for dataload is described in the *Reply Format* section of the Dataload documentation (5.18.5, p. 175)

The dataload SOAP API is essentially a wrapper around the dataload XML API. It allows you to pass in multiple `Item` objects to the function.

5.20.10 SOAP API admin Reference

login

Parameters:

- `username` - the user being logged in
- `password` - the password for the user

Returns:

- `authToken` - an authentication token for use in further requests

`login` logs you in to the appliance, supplying you with an authentication token that will be included in all further requests to show that you're logged in. All other Admin SOAP API calls require an authToken for use. It's sent in further requests via a SOAP Header.

listProfiles

Parameters:

- *none*

Returns:

- `ProfileName` - an array of profile names requests

Returns a list of all profiles that currently exist on the appliance. If no profiles exist, a successful response with no `ProfileNames` is returned.

getDocumentUsageOverview

Parameters:

- `Force` - whether to force recalculating the counts (see "Caching and Forcing" below)

Returns:

- `Timestamp` - When this count was performed
- `Timediff` - Number of seconds since this count was performed (see "Caching and Forcing" below)
- `CountTotal` - Total number of documents used by all profiles visible to your account
- `CountLimit` - Document count limit set by your product's license limit. Set to `0` if unlimited.
- `ProfileCount` - provides the document count for a single profile. The profile's name is provided in the `name` attribute.

getDocumentUsageOverview provides information about how many documents have been indexed by each of your profiles, and how the total compares to your product's license limit.

If there is a `New` walk occurring in a profile that already has live search data, then the larger of the two document counts is used.

Caching and Forcing

The Document Usage Overview is generated when requested, and then cached. If another request comes within 60 seconds, it will use the same cached information instead of re-calculating every one of the profiles' counts.

If you just performed some action that you know will cause a change in the document count, such as deleting a profile, you can override this behavior and force it to recalculate the totals. If you set `Force` to `Y` in the request, the document usage information will be recalculated regardless of when it was previously calculated and cached.

getProfileStatus

Parameters:

- `Profile` - name of the profile

Returns:

- `IsRunning` - whether or not the walk is currently running, set to `true` or `false`.

Returns information about the profile, currently just whether or not the profile is running.

addProfile

Parameters:

- `Profile` - name of the new profile
- `Type` *(optional)* - the type of profile, `standard` or `metasearch`. Defaults to `standard`.
- `CopyOf` *(optional)* - name of the profile to copy
- `ParametricField` - *unused in Webinator*
- `PrimaryKey` - *unused in Webinator*
- `Dataspace` - the directory to use for the profile's data

Returns:

- `Success` - will be set to `ok`, indicating the profile was created successfully.

Adds a new profile to Webinator. If there's any problem (already exists, invalid profile name, etc), a SOAP Fault will be thrown.

deleteProfile

Parameters:

- `Profile` - name of the profile to be deleted

Returns:

- `Success` - will be set to `ok`, indicating the profile was deleted successfully.

Deletes a profile from Webinator. If the profile didn't exist, the call will still succeed.

getSettings

Parameters:

- `Profile` - name of the profile. To get System Wide Settings, use `!SYSTEM`
- `Name` *(optional)* - array of setting names to get. If no `Name` is provided, all settings are returned.
- `TestOrLive` *(optional)* - whether to return the test settings, or live settings. Returns live settings by default.

Returns:

- `Setting` - an array of name/value pairs for the requested settings. A `TestOrLive` attribute on the settings indicate whether this applies to test, live, or both.

Gets a list of settings for the requested profile. You can request one or more specific settings by passing in `Name` parameters, or get all settings by not supplying a `Name`.

Some settings have test and live versions. You can request which version you'd like (defaults to live), and the returned settings indicate whether they apply to test or live. "Both" indicates that setting doesn't have different test and live versions.

setSettings

Parameters:

- `Profile` - name of the profile. To set System Wide Settings, use `!SYSTEM`
- `TestOrLive` *(optional)* - whether this should apply to test settings, live settings, or both. Defaults to both.
- `Setting` - multiple name/value pairs of settings that you'd like to set

Returns:

- `Success` - set to `ok`, indicating the settings were set properly.

Applies an array of settings for the given profile.

If there is any problem (such as an invalid setting name) in any one of the settings, a SOAP Fault is returned, and NONE of the settings are applied. This allows you to tweak the problem settings, and re-submit the entire batch again, without having them "partially applied" in between.

getQueryLogRaw

Parameters:

- `Profile` - name of the profile
- `Max` *(optional)* - maximum number of entries to return. Defaults to all
- `Skip` *(optional)* - number of entries to skip, used with `Max` to get all entries across multiple requests
- `DateFrom` *(optional)* - only return entries after the specified datetime
- `DateTo` *(optional)* - only return entries before the specified datetime

Returns:

- `Content` - A a tab-separated dump of the requested query log entries

Returns a tab-separated list of the profile's query log. The first line of `Content` describes the columns present.

pauseWalk

Parameters:

- `Profile` - name of the profile

Returns:

- `Success` - will be set to `ok`, indicating the profile will be paused

If a walk is running on this profile, this will pause the walk, build its indexes, and make the search live. It's the same effect as clicking "Pause walk and Live" on the Walk Status page.

stopWalk

Parameters:

- `Profile` - name of the profile

Returns:

- `Success` - will be set to `ok`, indicating the profile will be stopped

If a walk is running on this profile, this will stop and abandon the walk, leaving the existing search live. It's the same effect as clicking "STOP walk" on the Walk Status page.

startWalk

Parameters:

- `Profile` - name of the profile

Returns:

- `Success` - will be set to `ok`, indicating the walk was started successfully.

Starts a walk on this profile, the same effect as clicking "GO" in Basic Walk Settings.

getTask

Parameters:

- `Id` - id of the task to get

Returns:

- `TaskInfo` - a structure of information about the task
  - `Id` - id for the task, can be used in further requests
  - `Pid` - Process ID of the task, may be `-1` if task hasn't started
  - `Profile` - Profile for the task, or `!SYSTEM` for system-wide tasks
  - `Action` - the task's main action, defines its behavior (dispatch, updateindex, etc)
  - `Status` - the task's current status, possible values are:
    - `queued` - task is waiting to be run by the task monitor
    - `starting` - task has been launched and is starting up

    * `running` - task is running
    * `finished` - task completed successfully, see `Detail` may contain more information about its exit
    * `incomplete` - task exited before completing everything it could, `Detail` explains the reason for early exit (user canceling a walk, etc)
    * `died` - task ended abruptly and did not unregister itself, or the process was killed by another program.
- `Db` - notes which of the internal databases the task is operating on for the profile. May be `!NONE` for system wide tasks, or `!UNKNOWN` if the task is still starting and hasn't set a database yet.
- `DbRole` - currently unused
- `ParentType` - defines the type of parent recorded for launching this task. See `ParentData` below for values and their meaning.
- `ParentData` - Additional information about the process that launched this task, its content depends on the `ParentType`:
    * If `ParentType=task`, then `ParentData` contains the id of the task that launched this task.
    * If `ParentType=web`, then `ParentData` contains the name of the Webinator function that was invoked through the web to launch this task.
    * If `ParentType=cli`, then `ParentData` contains the name of the Webinator function that was invoked through the command line to launch this task.
    * If `ParentType=unknown`, then `ParentData` is the process id that launched this task.
- `Args` - The arguments that were used to launched this task, can occur multiple times for multiple arguments.
- `Started` - The dateTime this task was started. It will not be present for tasks that are queued or are still starting.
- `Updated` - The dateTime this task last updated its information. Most tasks should update regularly, although may go long periods of time without updating during long operations.
- `NextRun` - currently unused
- `Schedule` - currently unused
- `Description` *(optional)* - Additional information on what the task was launched to do. Some actions don't need a description (like `updateindex`), but others can provide additional information (`walk` lists its Base URL, etc).
- `Detail` - For running tasks, `Detail` may contain additional information about what the task is currently doing (building indexes, etc).

  For finished tasks, `Detail` may contain additional information about the reason the task exited.

  `ProgressInfo` - Some tasks provide additional progress information, such as walks or replication senders.
    * `Attempted` - the number of items attempted (pages attempted to be fetched, items sent for replication, etc).
    * `Saved` - the number of successful items
    * `Errors` - the number of items that encountered an error

    * `Bytes` - the number of bytes transferred or processed
    * `TodoCurrent` - the number of items to do at the current depth
    * `TodoNext` - the number of items to do at the next depth
    * `LoadAttempted1Min` - the average items per minute attempted, weighted towards the last minute
    * `LoadSaved1Min` - the average items per minute saved, weighted towards the last minute
    * `LoadAttempted15Min` - the average items per minute attempted, weighted towards the last 15 minutes
    * `LoadSaved15Min` - the average items per minute saved, weighted towards the last 15 minutes
    * `LoadAttempted1Hour` - the average items per minute attempted, weighted towards the last hour
    * `LoadSaved1Hour` - the average items per minute saved, weighted towards the last hour

`getTask` retrieves information about a specific task that is running or has been run. The id likely comes from a previous `getTasks` call, where checking again can be done more efficiently by giving the id exactly rather than the same criteria to `getTasks`.

getTasks

Parameters:

- `Profile` - the name of the profile to get tasks for, or `!SYSTEM` for system-wide tasks, or `!ALL` for all tasks.
- `Scope` *(optional)* - defines what tasks to get. Possible values are:
  - `running` - return only running tasks. This is the default when using `!SYSTEM` or `!ALL` for Profile.
  - `recent` - return tasks since the most recent walk or import. This is the default for individual profiles.
  - `!ALL` - all tasks, running and finished.
- `Db` *(optional)* - which profile database to retrieve tasks for. Possible values are:
  - `live` - the profile database currently being used for search.
  - `other` - if a `New` walk is running, this retrieves tasks for the walking database, not the live search database.

  The `Db` parameter is unused with Profile values of `!SYSTEM` and `!ALL`.
- `Max` *(optional)* - the maximum number of tasks to return, defaults to `-1` (unlimited).

Returns:

- `TaskInfo` - an array of structures of information about the task, see `getTask` for details.

`getTasks` retrieves information about tasks that are running or have been run. This can be used to keep tabs on what is running on a machine, or to tell when a specific profile has all of its tasks completed.

getThesauruses

Parameters:

- *none*

Returns:

- `Thesaurus` - an array of Thesaurus information
  - `Name` - name of the thesaurus
  - `Permutations` - what permutations apply to this thesaurus
  - `NumProfileUsing` - the number of profiles that are currently using this thesaurus

Returns information about all thesauruses that exist in Webinator.

setThesaurus

Parameters:

- `Name` - name of the thesaurus. If this thesaurus already exists, it will be replaced.
- `Permutations` - what permutations apply to this thesaurus. Possible values are `Full`, `Single`, or `None`. Defaults to `Single`.
- `Verbose` *(optional)* - If set to `Y`, verbose output of processing the thesaurus content will be included in the response.
- `Content` - the text content that should be used for the thesaurus. See the Thesaurus section for details on the format (5.3, p. 143).

Returns:

- `Output` - output of the thesaurus processing operation. Any errors will be listed in the text.

Creates or updates a thesaurus in Webinator. Once created, a thesaurus can be used in a profile by setting its `SSc_eqprefix` or `SSc_ueqprefix` to this thesaurus' `Name`.

deleteThesaurus

Parameters:

- `Name` - name of the thesaurus to delete

Returns:

- `Output` - output of the thesaurus deletion operation. Any errors will be listed in the text.

Deletes the thesaurus `Name`. Any profiles using this thesaurus will have their setting properly cleared.

5.20.11 Auth Proxy `conf/texis.ini` Section

The `conf/texis.ini` config file in the Texis install dir can have an optional **[Auth Proxy]** section with settings for the authProxy component of the Proxy Module. For details on the format of this file in general, see the "Texis Configuration File" section of the Texis manual. Settings in the **[Auth Proxy]** section are:

Auth Proxy Address
Default: unset
Defines what URL the appliance should use to authenticate its url results. The URL should point to the authProxy that is installed on this machine. Example:
`http://proxyMachine.example.com/authProxy`

Auth Proxy Disabled
Default: `false`
Can be set to `true` to disable the entire authenticating aspect of the Proxy Module. This is only necessary if, for some reason, you want the virtual directory for the Proxy Module to be authenticated, but you don't want to trigger the Auth Proxy behavior.

Auth Proxy Pipe
Default: `true`
Can be used to disable the Proxy Module's communcation with the Auth Proxy. You can set this to `false` if you're using some other sort of authentication other than the Auth Proxy. Most users should leave this unset or `true`.

Chapter 6

Reference

6.1 REX Syntax

The following sections discuss REX syntax, which is used by some settings (e.g. **Exclusion REX** p. 60, **Data from Field** p. 62) for searching and/or replacing text.

6.1.1 Expressions

- REX search expressions are composed of characters and operators. Operators are characters with special meaning to REX. The following characters have special meaning: "`\=+*?{},[]^$.-!`" and must be escaped with a "`\`" if they are meant to be taken literally. The string "`>>`" is also special and if it is to be matched, it should be written "`\>>`". Not all of these characters are special all the time; if an entire string is to be escaped so it will be interpreted literally, only the characters "`\=?+*{[^$.!>`" need be escaped.

- A "`\`" followed by an "`R`" or an "`I`" means to begin respecting or ignoring alphabetic case distinction. (Ignoring case is the default.) These switches stay in effect until the end of the subexpression. They *do not* apply to characters inside range brackets.

- A "`\`" followed by an "`L`" indicates that the characters following are to be taken literally, case-sensitive, up to the next "`\L`". The purpose of this operation is to remove the special meanings from characters.

- A subexpression following "`\F`" (followed by) or "`\P`" (preceded by) can be used to root the rest of an expression to which it is tied. It means to look for the rest of the expression "as long as followed by ..." or "as long as preceded by ..." the subexpression following the `\F` or `\P`. Subexpressions before and including one with `\P`, and subexpressions after and including one with `\F`, will be considered excluded from the located expression itself.

- A "`\`" followed by one of the following `C` language character classes matches any character in that class: `alpha`, `upper`, `lower`, `digit`, `xdigit`, `alnum`, `space`, `punct`, `print`, `graph`, `cntrl`, `ascii`. Note that the definition of these classes may be affected by the current locale.

- A "\" followed by one of the following special characters will assume the following meaning: `n`=newline, `t`=tab, `v`=vertical tab, `b`=backspace, `r`=carriage return, `f`=form feed, `0`=the null character.
- A "\" followed by `Xn` or `Xnn` where `n` is a hexadecimal digit will match that character.
- A "\" followed by any single character (not one of the above) matches that character. Escaping a character that is not a special escape is not recommended, as the expression could change meaning if the character becomes an escape in a future release.
- The character "^" placed anywhere in an expression (except after a "`[`") matches the beginning of a line (same as `\x0A` in Unix or `\x0D\x0A` in Windows).
- The character "`$`" placed anywhere in an expression matches the end of a line (`\x0A` in Unix, `\x0D\x0A` in Windows).

  Note: The beginning of line ("^") and end of line ("`$`") notation expressions for Windows are both identified as a 2 character notation; i.e., REX under Windows matches "`\x0D\x0A`" (carriage return, line feed) as beginning and end of line, rather than "`\x0A`" as beginning, and "`\x0D`" as end.
- The character "`.`" matches any character.
- A single character not having special meaning matches that character.
- A string enclosed in brackets ("`[]`") is a set, and matches any single character from the string. Ranges of ASCII character codes may be abbreviated with a dash, as in "`[a-z]`" or "`[0-9]`". A "^" occurring as the first character of the set will invert the meaning of the set, i.e. any character *not* in the set will match instead. A literal "–" must be preceded by a "\". The case of alphabetic characters is always respected within brackets.

  A double-dash ("`--`") may be used inside a bracketed set to subtract characters from the set; e.g. "`[\alpha--x]`" for all alphabetic characters except "`x`". The left-hand side of a set subtraction must be a range, character class, or another set subtraction. The right-hand side of a set subtraction must be a range, character class, or a single character. Set subtraction groups left-to-right. The range operator "–" has precedence over set subtraction. Set subtraction was added in Texis version 6.
- The "`>>`" operator in the first position of a fixed expression will force REX to use that expression as the "root" expression off which the other fixed expressions are matched. This operator overrides one of the optimizers in REX. This operator can be quite handy if you are trying to match an expression with a "`!`" operator or if you are matching an item that is surrounded by other items. For example: "`x+>>y+z+`" would force REX to find the "`y`"s first then go backwards and forwards for the leading "`x`"s and trailing "`z`"s.
- Normally, an empty expression such as "=" (i.e. 1 occurrence of nothing) is meaningless. However, if such an empty expression is the first or last in the list, and is the root expression (i.e. contains "`>>`"), it will constrain the whole expression list to only match at the start or end of the buffer. For example: "`>>=first`" would only match the string "`first`" if it occurs at the start of the search buffer. Similarly, "`last=>>=`" would only match "`last`" at the end of the buffer.
- The "`!`" character in the first position of an expression means that it is *not* to match the following fixed expression. For example: "`start=!finish+`" would match the word "`start`" and anything

past it up to (but not including the word "finish". Usually operations involving the NOT operator involve knowing what direction the pattern is being matched in. In these cases the ">>" operator comes in handy. If the '>>" operator is used, it comes before the "!". For example: ">>start=!finish+finish" would match anything that began with "start" and ended with "finish". *The NOT operator cannot be used by itself* in an expression, or as the root expression in a compound expression.

Note that "!" expressions match a character at a time, so their repetition operators count characters, not expression-lengths as with normal expressions. E.g. "!finish{2,4}" matches 2 to 4 characters, whereas "finish{2,4}" matches 2 to 4 times the length of "finish".

6.1.2 Repetition Operators

- A REX expression may be followed by a repetition operator in order to indicate the number of times it may be repeated.

  Note: Under Windows the operation "{X,Y}" has the syntax "{X-Y}" because Windows will not accept the comma on a command line. Also, N occurrences of an expression implies infinite repetitions but in this program N represents the quantity 32768 which should be a more than adequate substitute in real world text.

- An expression followed by the operator "{X,Y}" indicates that from X to Y occurrences of the expression are to be located. This notation may take on several forms: "{X}" means X occurrences of the expression, "{X,}" means from X to N occurrences of the expression, and "{,Y}" means from 0 (no occurrences) to Y occurrences of the expression.

- The "?" operator is a synonym for the operation "{0,1}". Read as: "Zero or one occurrence."

- The "*" operator is a synonym for the operation "{0,}". Read as: "Zero or more occurrences."

- The "+" operator is a synonym for the operation "{1,}". Read as: "One or more occurrences."

- The "=" operator is a synonym for the operation "{1}". Read as: "One occurrence."

6.1.3 RE2 Syntax

In Texis version 7.06 and later, on most platforms the search expression may be given in RE2 syntax instead of REX. RE2 is a Perl-compatible regular expression library whose syntax may be more familiar to Unix users than Texis' REX syntax. An RE2 expression in REX is indicated by prefixing the expression with "\<re2\>". E.g. "\<re2\>\w+" would search for one or more word characters, as "\w" means word character in RE2, but not REX.

REX syntax can also be indicated in an expression by prefixing it with "\<rex\>". Since the default syntax is already REX, this flag is not normally needed; it is primarily useful in circumstances where the syntax has already been changed to RE2, but outside of the expression – this should never be the case for Webinator REX expressions.

Note that while the \<re2\> and \<rex\> escapes are supported on all platforms, an RE2 expression itself may not be. Where unsupported, attempting to invoke an RE2 expression will result in the error

message "REX: RE2 not supported on this platform". (Windows, Linux 2.6 and later versions except i686-unknown-linux2.6.17-64-32 are supported.)

RE2 syntax is documented at https://github.com/google/re2/wiki/Syntax.

6.1.4 \¡nomatch\¿ Syntax

In Texis version 7.07.1584374000 20200316 and later, the escape \<nomatch\> may be given as the sole contents of a REX expression. This will match and return non-empty data that is not returned by any other (non-\<nomatch\>) expression. Since it is a negation, it may only be given if other (non-\<nomatch\>) expressions are given as well, e.g. with <rex> in Vortex. This may be useful when parsing text with multiple complex expressions, as a catch-all to match remaining text/space etc. that "falls through".

6.1.5 REX Caveats and Commentary

REX is a highly optimized pattern recognition tool that has been modeled after the grep and lex Unix family of Unix tools. Wherever possible REX's syntax has been held consistent with these tools, but there are several major departures that may bite those who are used to using the grep family.

REX uses a combination of techniques that allow it to operate at a much faster rate than similar expression matching tools. Unlike grep, Rex is both deterministic and non-directional. This may cause some initial problems with users familiar with grep's way of thinking.

REX always applies repetition operators to the longest preceding expression. It does this so that it can maximize the benefits of using its rapid state skipping pattern matcher.

If you were to give grep the expression: "ab*de+"

It would interpret it as: an "a" then 0 or more "b"s then a "d" then 1 or more "e"s.

REX will interpret this as: 0 or more occurrences of "ab" followed by 1 or more occurrences of "de".

The second technique that provides REX with a speed advantage is ability to locate patterns both forwards and backwards indiscriminately.

Given the expression: "abc*def", the pattern matcher is looking for "Zero to N occurrences of 'abc' followed by a 'def'".

The following text examples would be matched by this expression:

```
abcabcabcabcdef
def
abcdef
```

But consider these patterns if they were embedded within a body of text:

```
My country 'tis of abcabcabcabcdef sweet land of def, abcdef.
```

A normal pattern matching scheme would begin looking for "abc*" . Since "abc*" is matched by every position within the text, the normal pattern matcher would plod along checking for "abc*" and then whether it's there or not it would try to match "def". REX examines the expression in search of the the most efficient fixed length subpattern and uses it as the root of search rather than the first subexpression. So, in the example above, REX would not begin searching for "abc*" until it has located a "def".

There are many other techniques used in REX to improve the rate at which it searches for patterns, but these should have no effect on the way in which you specify an expression.

The three rules that will cause the most problems to experienced grep users are:

1. Repetition operators are always applied to the longest preceding fixed length expression.
2. There must be at least one subexpression that has one or more repetitions.
3. No matched subexpression will be located as part of another.

Rule 1 Example : "abc=def*" means one "abc" followed by 0 or more "def"s.

Rule 2 Example : "abc*def*" *cannot* be located because it matches every position within the text.

Rule 3 Example : "a+ab" is idiosyncratic because "a+" is a subpart of "ab".

6.1.6 Some Useful REX Expressions

- To locate phone numbers:

```
1?\space?(?\digit\digit\digit?)?[\-\space]?\digit{3}-=\digit{4}
```

- To locate social security numbers:

```
\digit{3}-=\digit{2}-=\digit{4}
```

- To locate text between parentheses:

```
(=[^()]+)       <- without direction specification
     or
>>(=!)+)        <- with direction specification
```

- To locate paragraphs delimited by an empty line and 4 spaces:

```
>>\n\n=\space\P{4}!\n\n\space\space\space\space+\F\n\n=\space{4}
```

- To locate numbers in scientific notation; e.g., "-3.14 e -21":

```
[+\-]?\space?>>[0-9]+\.?[0-9]*\space?e?\space?[+\-]?\space?[0-9]+
```

6.2 REX Replace Syntax

When replacing the match of a REX/RE2 expression, the replacement string has the following syntax:

- The characters "`?#{}+\`" are special. To use them literally, precede them with the escapement character "`\`'.
- Replacement strings may just be a literal string or they may include the "ditto" character "`?`'. The ditto character will copy the character from the search buffer that is in the same position as the ditto character is in the replacement string.
- A decimal digit placed within curly-braces (e.g. `{5}`) will copy the character at that index (of the search buffer) to the output. Characters are indexed starting at 1. An index beyond the end of the search buffer will not print anything.
- A "`\`" followed by a decimal number will copy that subexpression (REX) or parenthetical numbered capturing group (RE2) to the output. Subexpressions and groups are numbered starting at 1. Named groups (RE2) are not currently supported. See p. 195 for more on RE2.
- The sequence "`\&`" will copy the entire expression match (sans `\P` and `\F` portions, if REX syntax) to the output. This escape was added in Texis version 7.06.
- A plus-character "+" will place an incrementing decimal number to the output. One purpose of this operator is to number lines.
- A "`#`" followed by a number will cause the numbered subexpression (REX) or parenthetical numbered capturing group (RE2) to be printed in hexadecimal form. Subexpressions and groups are numbered starting at 1. Named groups (RE2) are not currently supported.
- Any character in the replace-string may be represented by the hexadecimal value of that character using the following syntax: `\x`hh where hh is the hexadecimal value.

6.3 Supported File Formats

- Adobe Acrobat - `.pdf` (Versions 1-10)
- Ami Professional - `.sam` (Versions 1.0-3.1)
- ASCII - varies (Any plain text format. MIME type `text/plain`)
- Atom feeds `.atom, .xml`
- Azure Blob Listings - `.xml`
- Bzip2 - `.bz2` (Decompress and process supported sub-files)
- CCMail (All versions)
- Compress - `.Z` (Decompress and process supported sub-files)

- CTOS DEF
- DG CEOWrite(3.0)
- dBase - `.dbf` (All versions)
- DCIMARC
- DEC WPS-Plus - `.wpl` (through 4.1)
- Enable - `.wpf` (1.0 through 2.15)
- FoxPro - `.dbf` (dBase workalike)
- FrameWork - (III 1.0, 1.1, IV)
- GIF - `.gif` (textual meta data only)
- Gzip - `.gz` (Decompress and process supported sub-files)
- FrameWork - (III 1.0, 1.1, IV)
- Harris Typesetter
- HTML pages - `.htm .html .php .asp .cfm` etc. (All versions)
- IBM Writing Assistant - `.iwa, .wrt` (All versions)
- Interleaf (5.2, 6.0)
- JPEG images- `.jpg, .jpeg`
- Legacy - `.leg` (1.x, 2.0)
- Lotus 1-2-3 - **.wks .wk1 .wk2 .wk3 .wk4** (Versions 1A, 2.0 through 5.0)
- MacWrite II - `.mcw .mw` (1.0, 1.1)
- MacWrite Pro - `.mcw .mw` (1.0)
- MS Excel Spreadsheets - `.xls .xlsx xltx`
- MS Help files - `.hlp` (only on Windows with ”helpdeco”)
- MS Internet Explorer Save-as Files - `.mht`
- MS CHM files - `.chm` (only on Windows with ”hh”)
- MS Office
- MS Outlook emails - `.msg .eml` (All versions)
- MS Powerpoint presentation - `.ppt .pptx .potx`(through 2007)
- MS Transport Neutral Encoding Format - `.tnef` (All versions)

- MS Word Documents - `.doc .docx .dotx`
- MS Write - `.wri` (3.x)
- OfficePower - `.op6 .op7` (6.0, 7.0)
- OfficeWriter - `.ow4 .ow5 .ow6` (4.0, 5.0, 6.0, 6.1)
- Open Document - `.odt .ods` etc.
- PeachText 5000 - `.pea` (Version 2.12)
- PFS: First Choice - `.pfs` (1.0-3.0)
- PFS: Write - `.pfs` (Version C)
- Plain text - `.txt` (All versions)
- PostScript - `.ps` (All versions)
- Professional Write - `.pw` (1.0, 2.1, 2.2)
- Professional Write Plus - `.pw .pwp` (1.0)
- Q&A for DOS - `.qa .qw .dtf` (2.0)
- Q&A Write for Windows - `.dtf` (3.0)
- Rapidfile Memo Writer - `.mmo` (1.0, 1.2)
- Rar - `.rar` (Decompress and process supported sub-files)
- RFC882 Mail (All versions)
- RSS Feeds `.rss .xml`
- SGML files - `.sgml` (All versions)
- Shockwave/Flash - `.swf` (All versions)
- Tagged Image File Format (TIFF) - `.tif .tiff` (Meta data only)
- Tar - `.tar` (Extract and process supported sub-files)
- Total Word - `.tw` (1.2, 1.3)
- Uniplex onGo (v7)
- Usenet News
- Vines Mail
- Volkswriter - `.vw .vw3` (3.0, 4.0)
- Wang WITA - `.iwp` (through 2.6)

- Wiziword - `.doc` (All versions)
- Wordpad document - `.rtf` (All versions)
- Word Perfect - `.wpd` (4.1 through 6.1, +Mail Merge)
- Word Perfect Mac (1.0 through 3.1)
- Wordstar - `.ws .wsd` (3.3 through 7.0)
- Wordstar 2000 - `.ws2` (3.0, 3.5)
- WriteNow - `.wn` (3.0)
- XML - `.xml` (Applies XSL if present, otherwise treats as HTML)
- XyWrite - `.xy .xy3 .xy4` (III, III Plus, IV)
- XyWrite for Windows - `.xyw` (4.0)
- Zip - `.zip` (Decompress and process supported sub-files)

6.4 Database and File Usage

Webinator maintains a database that contains text from HTML pages, links to other pages, and a list of categories.

When the Webinator walker runs it creates a new database, under your specified data directory, to hold the new walk. It then dispatches a separate process for each web site it needs to visit and another to handle all of the "Single Pages". Each of these retrieves all of the pages in its base list and stores the text of the HTML page to the html table and the hyperlinks to the refs table. All of the desirable URLs from the page that have not been seen before are placed into an internal "todo" list. After all of the base URLs are processed the process repeats with the internal todo list. When there's nothing left in the todo list processing is complete.

Once all of the walking is complete the indices needed for searching are created on the data. Then the new database is flagged as the "live" one and the old database is deleted. Therefore your disk must have sufficient space for 2 complete databases plus temporary space used during the indexing step.

The databases are stored under your specified data directory. The databases are called `db1` and `db2`. Webinator alternates between using these two names.

Note that the above applies to a walk type of `New`. During a walk type of `Refresh` only one database, the "live" one, is used.

Webinator also maintains a file containing the detailed report for each walk. This file has the same name as the database with `.long` appended to the end. Also, a single file called `summary` is maintained with short summary information about the state of the database.

Given a data directory named `.../default` there may also be the following:

`.../default/db1` an actual walk database

`.../default/db2` an actual walk database

`.../default/db1.long` detailed walk report. Displayed when viewing `Walk Status`

`.../default/db2.long` detailed walk report. Displayed when viewing `Walk Status`

`.../default/summary` summary walk report. Displayed as `Walk summary` when viewing `Walk Settings`

Webinator, being based on Texis, also has the notion of a global "default" database. This database resides in the installation directory. On Unix it is called $INSTALLDIR$`/texis/testdb`. On Windows it is called $INSTALLDIR$`\texis\testdb`. This database is used to hold all of the profile and account settings. It does not contain any walked data. It is recommended that you *not* use this as your data directory.

Each setting has a record in the `options` table of the default database. See section 6.6 (p. 205) for the list of fields in the table. At each complete rewalk the current options settings are copied into an options table in the walk database. These options are not changed as settings are modified and are not otherwise used unless a search is performed setting the database with `db` instead of setting the profile with `pr`.

6.5 Walk Database Tables and Fields

Table 6.1: Fields in `html` table

Field	Description
`id`	Unique record id
`Hash`	Document hash for duplicate content detection
`Size`	Size of retrieved raw document (i.e. HTML)
`Visited`	The date the page was modified (or fetched if modified not set)
`Dlsecs`	The number of seconds needed to fetch the page
`Depth`	The number of URLs traversed to reach the page
`Url`	The URL of the real HTML page
`Title`	The title of the page
`Body`	The formatted textual content of the page, in Storage Charset (UTF-8)
`Keywords`	The `keywords` meta data from the page
`Description`	The `description` meta data from the page
`Meta`	Other meta data from the page, separated by newlines
`Catno`	List of categories to which the URL belongs
`CatnoLowest`	Lowest `Catno` value
`Modified`	The date the page was last modified
`NextCheck`	The date the page should next be refreshed
`Views`	The number of times this URL has been viewed (shown in results)
`Clicks`	The number of times this URL has been clicked (in results)
`CTR`	Click-through ratio
`Pop`	Popularity (number of pages linking to this page)
`MimeType`	MIME type of original page
`Charset`	Character set of page as stored (usually Storage Charset)

Table 6.2: Fields in `refs` table

Field	Description
`Url`	The URL of the HTML page
`Ref`	The URL of a reference (link) on the HTML page

Table 6.3: Fields in `categories` table

Field	Description
`Catno`	The number for the category
`OverlapsLower`	Y if some member(s) also in a lower category
`Url`	The URL pattern for the category
`Category`	The name of the category

Table 6.4: Fields in `error` table

Field	Description
`Url`	The URL of an HTML page that could not be retrieved
`Reason`	The reason it could not be retrieved
`id`	Unique record id (includes timestamp info).

Table 6.5: Fields in `querylog` table (if query logging enabled)

Field	Description
`id`	Contains the date and time of the query (unique record id)
`Client`	The hostname of the web client that performed the query
`Query`	The user's query as entered

6.6 Options Table Fields

These are the options table fields (maintained in the default database):

Table 6.6: Fields in `options` table

Field	Description
`id`	Unique id for the record
`Profile`	The name of the profile that the record belongs to
`Name`	The name of the setting
`Type`	The data type of the setting (always `String`)
`String`	The value of the setting
`Int`	Unused
`Float`	Unused
`Strlist`	Unused

You can look at the `SYSCOLUMNS` and `SYSINDEX` tables of the database for details about the field types, sizes, and indices.

6.7 Customizing the Search

You may make common changes to Webinator's search appearance by using `Search Settings` from the administrative interface main menu. But you are not limited to those features. You may change any and all aspects of the search program's appearance and behavior by modifying the supplied `search` script or writing an altogether new one.

For details on programming with Texis Web Script (Vortex), see the manual at the Thunderstone web site, `http://docs.thunderstone.com/site/vortexman/`.

The following section describes some important points about the internals of the default search script that comes with Webinator. The search script is fairly heavily commented to aid in finding your way around within it.

The `init` function is called from every entry point. It is a good place to put settings that should always (or often) apply. This function understands the old (version 2) style specification of database by the `db` variable as well as the current method of extracting the database name from the profile named by the `pr` variable.

The `top` function displays the common HTML for the beginning of every page generated by the search script. This does not include the search form. This function is where you would place styles and navigation menus.

The `bottom` function is the complement to the `top` function. It displays the common HTML footer for the end of every page.

The `showform` function displays the search form with all current settings indicated.

The `qpar` and `fpar` functions process the user's form submission and apply appropriate search settings.

The `credit` function displays the Thunderstone credit on the search results. This is required for free users but may be changed or emptied for paid users.

The `result` function is called for each matching record to display. It then calls the configured `result*` function to generate the desired output style.

The `mlt` function is called to setup the search when the end user selects "`Find Similar`" (aka More Like This).

The `similar` function may be called directly to find pages within the database that are similar to the content of the URL specified. It has the same concept of "`Find Similar`" but will work on any specified URL, not just those displayed as the result of a search. It would be invoked something like this on any HTML page.

```
<a href="/cgi-bin/texis/webinator/search/similar.html?↪
   ↪pr=default&ref=http://example.com/somepage.html">↪
   ↪Find pages similar to somepage.html</a>
```

or

```
<a href="/scripts/texis.exe/webinator/search/similar.html?↪
   ↪pr=default&ref=http://example.com/somepage.html">↪
   ↪Find pages similar to somepage.html</a>
```

Set `default` above to the search profile you're using.

It will lookup that URL in the database or, if it's not in the database, fetch it from the webserver. It will then search the database looking for indexed pages similar to the specified page.

The `main` function is the standard Vortex default entry point. This is the function that is first called when users click "Submit" on the search form.

The `search` function does the core work of finding matching documents within the database. It calls `showform` and `qpar` then starts searching. For every match the `result` function is called. The `summary` function is called before the first match is displayed to display the search results summary. It is called again at the end of the results list.

The `putmsg` function handles errors that may occur and displays them in a somewhat more user friendly fashion. See the Vortex manual for details about how `putmsg` is used to capture errors.

6.8 Customizing the Walker

You may make many changes to Webinator's walk behavior by using `Walk Settings` from the administrative interface main menu.

But you are not limited to these features. You may change any and all aspects of the walker's behavior by modifying the supplied `dowalk` script. (The `webinatoradmin` script supplied with version 4 and earlier releases has been combined into `dowalk` for atomicity.)

For details on programming with Texis Web Script (Vortex), see the manual at the Thunderstone web site, `http://www.thunderstone.com/`.

The following describes some important points about the internals of the `dowalk` script that comes with Webinator. The `dowalk` script is fairly heavily commented to aid in finding your way around within it.

The `dowalk` script actually consists of 2 Vortex script files concatenated. The first part contains the walker/indexer and settings reading code. The second part of the file provides the management interface that is used from a web browser.

The `dispatch` function is the primary external entry point for performing a new walk. It load settings, sets up logging and databases, then invokes other processes in parallel (according to maximum servers setting). When all of the walking is complete it removes commonality from pages (if that option is set), creates the indices needed for searching the database, then makes the new database live and deletes the old database.

The `stop` function is an external entry point that is used to signal (using `<loguser>`) a walk that is in progress that it should stop. The walkers check for this signal (using `<userstats>`) at various points and will quit when it is detected.

The `reindex` function is an external entry point that is used to drop and recreate the Metamorph index on the html table. This is needed after changing the word definition expressions.

The `remakeindex` function is an external entry point that is used to drop and recreate all indices on the database. It it only for use if one or more non-Metamorph indices get corrupted by disk errors or such.

The `recat` function is an external entry point that is used to recategorize the html table based on the

current (presumably changed) categories (p. 55). This may take some time on large walks.

The `ifmodified` function is an external entry point that is used to tell the dispatcher to run only if chkneedwalk indicates a walk is needed.

The `usage` function is called when you invoke `dowalk` incorrectly and prints a terse summary or correct usage options.

The `doplugin` function handles files that are not HTML or text, such as PDF and MSWord. It determines the correct options for `anytotx` based on the fetched page's MIME type or extension. It then calls the `dofilt` function which actually runs `anytotx` to perform the conversion to text and the extraction of meta information such as Title. It will make up a title for the document if none is returned by `anytotx`.

The `settings` function calls the defaults, readsettings, and applysettings functions, in order. This function is called by most entry points to get default and current settings for a given profile before proceeding with any work.

The `updatemmindex` function is called (sometime after having called settings) to create or update the Metamorph index on the html table.

The `maketables` function is called (sometime after having called settings) to create all of the Webinator tables. This function does nothing for Webinator-only licenses. For Webinator-only licenses the tables are created automatically by Texis when the database is created. The schema may not be changed.

The `walk` function is the core which walks all desired URLs on a single site. It always processes breadth first (i.e. it gets all URLs at a given depth before proceeding to the next level down). Any desired URLs that reside on a different site are placed into the database's todo table for processing by the dispatcher.

The `fetchset` function is used in various places to fetch one or more URLs (using the maximum threads setting) simultaneously.

The `manglepage` function is called before extracting text and hyperlinks from an HTML page. It allows the page to be modified before processing. This is where the ignore/keep tags are handled.

The `getrobotstxt` function fetches the robots.txt file from a given site and checks for any exclusions for Webinator. These exclusions are later added to the list of URL rejection patterns.

The `chkneedwalk` function is called to check if a rewalk is required. It fetches the page to see if the modification date has changed. Or, if the web server does not provide a modification date it compares the content to what it was previously. It sets an internal flag if a rewalk is needed.

The `putmsg` function intercepts error messages to provide special handling for some, and recording of most.

The `go` function is an external entry point used by the dispatcher when it starts up child processes to walk a specific site or set of URLs.

The `singles` function is an external entry point that is used to fetch all of the single page URL. It is called by the dispatcher as the first parallel process. Therefore single pages will generally be fetched earliest in a new walk.

The `rmlocks` function is used to remove any stale locks and monitor processes on a database and dismantle the locking structure. This is done before physically removing a database from the system.

The `geturl` function is a utility function that may be used to find out what the walker will think about a given URL using the current walk settings. It is invoked as follows:

```
texis profile=PROFILE top=THEURL dowalk/geturl.txt
```

This can generate a lot of output for a page of any size so you may want redirect it to a file that you can examine with your favorite viewer/editor.

```
texis profile=PROFILE top=THEURL dowalk/geturl.txt >FILE.txt
```

The `getrobots` function is a utility function that may be used to find out what the walker will think about a given robots.txt using the current walk settings. It is invoked as follows:

```
texis profile=PROFILE top=THEURL dowalk/getrobots.txt
```

This can generate a lot of output for a page of any size so you may want redirect it to a file that you can examine with your favorite viewer/editor.

```
texis profile=PROFILE top=THEURL dowalk/getrobots.txt >FILE.txt
```

6.9 Texis ISAPI

6.9.1 Overview

Texis ISAPI uses IIS's Internet Server API (ISAPI) to allow Webinator on IIS to use a Unix-style web address (no `.exe` in the path). Many URL-scanning programs see having `.exe` in the address path as an indication of an attempted exploit, so removing this can be desirable.

The other advantage is that Texis ISAPI can be installed on an IIS machine different from the machine that Webinator is installed on. Webinator can be installed on a dedicated machine inside your intranet, and your IIS web server can use ISAPI to display its content.

Installation and usage of Texis ISAPI does not in any way prevent usage of the conventional CGI method. Both can be used simultaneously, if desired.

6.9.2 How it Works

The Texis ISAPI software acts as a pass-through. IIS is configured to give requests to Texis ISAPI, which in turn passes it along to Webinator. Texis ISAPI receives Webinator's response, and passes that response back to the web browser.

There are two types of ISAPI programs: filters and extensions. Texis ISAPI contains both a filter and an extension in `ProxyModule.dll`, although you will only use one or the other (which one depends on

which version of IIS you're running). Both the Texis ISAPI filter and Texis ISAPI extension offer the same features and functionality, they only differ in how they are implemented in and communicate with IIS (the installer is able to set up either for you automatically).

- **IIS 5 or earlier**
  In IIS 5 or earlier, an ISAPI Filter is used. This is installed as a global filter, as is required of all filters that use `SF_NOTIFY_READ_RAW_DATA`. It is invoked for every request, but takes no action unless the request begins with `/texis`. If the request does, Texis ISAPI takes control of the request and processes it appropriately.
- **IIS 6 or later**
  On IIS 6 or later, `SF_NOTIFY_READ_RAW_DATA` for filters is explicitly denied, so Texis ISAPI uses an ISAPI Extension mapped as a Wildcard Application Mapping. The installer accesses the default site and creates a new virtual directory, `texis`. It creates custom Application Settings for that virtual directory, and adds Texis ISAPI's DLL file as a Wildcard Application Map. This means that any request that comes to that virtual directory (i.e. `/texis/...`) will not map to a real file location, but will instead be handed off to Texis ISAPI. The installer also adds Texis ISAPI as an "allowed" extension to the IIS Web Service Extensions Restriction List.

  Wildcard Application Maps are not available prior to IIS 6, so the filter is still used for earlier versions.

6.9.3 Settings for Texis ISAPI

Texis ISAPI must be configured for how to contact Webinator. It needs a port number and a host (if it's not the localhost). Regardless of whether loading settings was successful or not, an entry will be made in the `Application` Event Log detailing either what settings were loaded, or why settings couldn't be loaded.

Texis ISAPI will first attempt to read a port number from a locally installed Webinator's texis.ini.

Reading values from `texis.ini`

Texis ISAPI does this by first looking in the registry for an `InstallDir` value in the `HKEY_LOCAL_MACHINE\Software\Thunderstone Software` key to locate the installation path.

It tries to read the following values from the `[Httpd]` section of `texis.ini`:

`Port`: This setting serves double-duty: it tells monitor web server what port it should listen on, and it tells Texis ISAPI what port it should use to connect to the monitor web server. The default `10700` should be fine.

If it is unsuccessful in reading values from `texis.ini`, it will then attempt to read values from the registry.

Reading values from the Registry

Texis ISAPI will attempt to read the following values from the registry key `HKEY_LOCAL_MACHINE\Thunderstone Software\ISAPI`:

`port` (DWORD): the port that Texis ISAPI should use to connect to the remote monitor web server.

`host` (String): the hostname of the webinator machine that Texis ISAPI should use. If no hostname is found, `localhost` is assumed.

If Texis ISAPI is unable to read a `port` value from the registry, then it will disable itself and makes an Event Log entry detailing why it couldn't read from texis.ini and why it couldn't read from the registry.

6.10 CGI Mapping by Vortex File Extension

The following sections detail how to manually configure some web servers to run Webinator's Vortex scripts by URL file extension (e.g. `.vs` or `.vsc`), instead of by URL directory (e.g. `/cgi-bin/texis` or `/scripts/texis.exe`). This will allow[1] URLs such as `/webinator/dowalk.vs` to be run, instead of `/scripts/texis.exe/webinator/dowalk.vs`.

Notes:

- Windows: if the Texis ISAPI filter/extension is being used (p. 209), this procedure may not be needed, nor does it affect those URLs.
- This procedure may already have been performed by the Webinator installation wizard (especially under Windows).
- This procedure requires Vortex version 6 or later, with the default configuration of `.vs` in `Vortex Source Extensions` in `texis.ini`. (Run `texis -version` from a command prompt to determine your version.)

6.10.1 Microsoft IIS

To manually configure Microsoft IIS to map Vortex scripts by file extension (`.vs`), Vortex version 6, and IIS version 6.0 or later are required. Earlier versions may not map extended-Vortex-syntax URLs (e.g. with `/func.html` appended) properly. Note that this procedure may already have been performed by the Webinator installation wizard. This procedure should be performed as an administrator.

1. **Open the IIS configuration:**
   (a) Right-click on `My Computer` on the desktop.
   (b) Select `Manage...` to open the `Computer Management` application.
   (c) Expand the `Services and Applications` tree (click on its plus-sign).
   (d) Expand the `Internet Information Services` tree.
2. **Add `texis.exe` to Web Service Extensions:**
   (a) Double-click on `Web Service Extensions`.

[1] Depending on the license obtained from Thunderstone; this generally requires at least a purchased license.

(b) If an item named `Texis` already exists under the `Web Service Extension` list, double-click it, and click the `Required Files` tab.
Otherwise, if an item named `Texis` does *not* already exist under the `Web Service Extension` list, click the `Add a new Web service extension...` link and enter `Texis` into the `Extension name:` box.

(c) Click `Add...` on the `Properties` or `New Web Service Extension` dialog.

(d) Enter the path to the `texis.exe` executable in the `Path to file:` box. This is your Webinator install directory plus `\texis.exe`, e.g. `C:\morph3\texis.exe` or `"C:\Program Files\Thunderstone Software\Webinator\texis.exe"`. Double-quote the path if it contains spaces.

(e) Click `OK` to close the `Add file` dialog box. The `texis.exe` path should now be listed under `Files:` as `Allowed`, or under `Required files:`.

(f) Click `OK` to close the `Web Service Extension Properties - Texis` or `New Web Service Extension` dialog box.

3. **Add a file-extension application mapping to the web site:**

(a) Expand the `Web Sites` tree (on the left).

(b) Right-click the appropriate web site.

(c) Select `Properties...` from the popup menu.

(d) Click the `Home Directory` tab (at top of dialog).

(e) Click the `Configuration...` button (lower right).

(f) Scroll through the `Application extensions` list. If an `Extension` for `.vs` (Vortex) already exists, double-click it. If not, click `Add...` to create one.

(g) In the `Executable:` box, enter the *identical* path to the `texis.exe` executable that you entered above.

(h) In the `Extension:` box, enter `.vs` to map Vortex files with that extension in URLs to the `texis.exe` executable.

(i) Check `All verbs` and `Script engine`.

(j) *Un*check `Verify that file exists`[2]

(k) Click `OK` to close the `Add/Edit Application Extension Mapping` dialog.

(l) Click `OK` to close the `Application Configuration` dialog.

(m) Click `OK` to close the `... Web Site Properties` dialog.

4. **Close the** `Computer Management` **application**

6.10.2 Apache

To manually configure the Apache web server to map Vortex scripts by file extension (`.vs`), either a redirect handler can be added to run Vortex scripts (preferred), or the scripts themselves can be directly executed (alternate). The redirect-handler method is preferable, as the latter method has some drawbacks.

[2] If `Verify that file exists` were checked, Vortex URLs would result in 404 errors, because IIS would not find the Vortex files in the web site's document root as it expects. They are in Vortex's `ScriptRoot` dir instead.

Preferred Method: Redirect Handler

To configure Apache to run Vortex scripts by file extension (`.vs`) using the preferred redirect-handler method, Vortex version 6, and Apache version 2.1 or later are required[3]. This procedure is intended for Unix systems and should be performed by `root`. It configures `texis` as a redirect handler to run Vortex scripts. Consult your web server manual (or online at `http://www.apache.org/`) for details and consequences on the directives used in this procedure:

1. Find the existing URL (not filesystem dir) on your server that runs `texis`. Typically this is `/cgi-bin/texis`. If there is no existing URL to run `texis`, consult your web server manual and configure the web server to do so. Typically this is done with a directive similar to:
   `ScriptAlias /cgi-bin/ /var/www/cgi-bin/`
   Note that you may also have to copy or symlink the `texis` executable to `/var/www/cgi-bin` (though this is typically already performed by the Webinator installation wizard).
2. Open the Apache configuration file (typically `/etc/httpd/conf/httpd.conf`; see your web server manual) with a text editor.
3. Add the following two lines, preferably near (but outside) the existing CGI directives:

   ```
   Action texis-vortex /cgi-bin/texis virtual
   AddHandler texis-vortex .vs
   ```

   If your existing configuration uses a URL other than `/cgi-bin/texis` to run `texis`, then change the `Action` directive appropriately. Note that the `Action` directive requires that it *must* be a URL, not filesystem, path. Note that Apache version 2.1 or later is needed for the `virtual` keyword; with 2.0, omit it.
4. Save the configuration file and exit the editor.
5. Re-start the web server (typically with `/etc/init.d/httpd restart`; check your web server manual).

Note that despite the `virtual` keyword in the `Action` directive, Apache may still require that the parent dir(s) of Vortex scripts exist in the document root (even though Vortex scripts are typically in Vortex's `ScriptRoot` dir, i.e. `texis/scripts` in the Webinator install dir). This is believed to be a bug in Apache; it was noted in version 2.2.4 at least. Therefore, you may have to create a `webinator` (or other) directory in your web server's document root (usually `/var/www/html`). This dir may have already been created by the Webinator installation wizard. If the parent dir(s) are missing in the document root, it may one potential cause of `404 Not Found` errors when attempting to run Vortex scripts by extension.

Alternate Method: Direct Execution

An alternate method for configuring Apache to run Vortex scripts by extension (`.vs`) is to have Apache execute the scripts directly, instead of using a redirect handler. This method is less desirable than the redirect-handler method (above) for several reasons:

[3] Apache version 2.0 may be used, but it does not support the `virtual` keyword for `Action`. Consequences are noted in the procedure.

- The Vortex scripts must exist in the web server document root, not Vortex's `ScriptRoot`. This means the document root must be writable by the `texis` or CGI user (for `.vsc` compilation), and the script sources are accessible to users (though it may be possible to configure the web server to prevent the latter).
- Every script must be edited to contain a `#!/usr/local/morph3/bin/texis` prefix at the start.
- Every script must be have its execute bits set via `chmod`.

However, this alternate procedure may be used if the preferred redirect-handler method is not possible or practical for some reason. Consult your web server manual (or online at `http://www.apache.org/`) for details and consequences on the directives used in this procedure:

1. Open the Apache configuration file (typically `/etc/httpd/conf/httpd.conf`; see your web server manual) with a text editor.
2. Find the `<Directory ...>` directive that applies to the same directory as the `DocumentRoot` directive. This is typically (but not always) `<Directory /var/www/html>`.
3. Inside this `<Directory ...>` directive, add `ExecCGI` as an option to the `Options` directive.
4. Outside the directive (i.e. the line after `</Directive>`), add this line:

   ```
   AddHandler cgi-script .vs
   ```

5. Save the configuration file and exit the editor.
6. Copy all Vortex scripts you wish to run from the Web by this method, from `ScriptRoot` to the same dir under document root. (This step is not needed if for some reason you have edited your Vortex `texis.ini` file and set `ScriptRoot` to `%DOCUMENT_ROOT%`.) For example, the `dowalk.vs` and `search.vs` scripts in `/usr/local/morph3/texis/scripts/webinator` should be copied to the `/var/www/html/webinator` directory (create it if needed), assuming your Apache `DocumentRoot` is `/var/www/html`.
7. Edit every script you copied in the previous step with a text editor, and add this line as the *very* first line in the file:

   ```
   #!/usr/local/morph3/bin/texis
   ```

   (If your Webinator install directory is not `/usr/local/morph3`, change it appropriately.) This step and the next are needed even if `ScriptRoot` is `%DOCUMENT_ROOT%`.
8. Run `chmod a+rx` on every script you copied and edited.
9. Re-start the web server (typically with `/etc/init.d/httpd restart`; check your web server manual).

Note that if you are running Apache for Windows, you may also need to set or edit the `ScriptInterpreterSource` directive; see your web server manual.

6.11 Third-Party Software

See the Vortex manual for a list of third-party software used by Webinator.

6.12 Version Differences

See the Vortex manual for a list of features and differences between major versions of Webinator.

Chapter 7

Search Interface Help

7.1 Forming a Query

Webinator's search can be as simple or as complex as you need it to be. Usually you will just need to enter a few words that best describe that which you are trying to locate. To perform more complicated searches you might use any combination of logic operators, special pattern matchers, concept expansion, or proximity operations.

Example: `nature conservation organization`

7.1.1 Query Rules of Thumb

- If you get too many junk or nonsense results, try:
  - Add some more words to your query.
  - Decrease the range of the `Proximity` control.
  - Change the `Word Forms` control to `Exact`.
  - Look at the Match Info and see why they are showing up.
  - Use the Exclusion Operator `(-)` to remove unwanted terms.
  - If you are searching for a phrase, hyphenate the words together.
- If you don't get any results, or just too few:
  - Remove some more words to your query.
  - Examine your spelling.
  - Increase the scope of the `Proximity` control.
  - It just might not be there?

7.1.2 Overview of Query Abilities

Webinator is based on Texis and as such it shares its text query abilities with all of Thunderstone's products. Throughout our documentation you will see references to Metamorph or Texis. This is because all of our products share a common text query language. This document provides only a brief overview of this language.

If you'd like to know more see the online manual at
`http://docs.thunderstone.com/site/texisman/link_mmq.html`.

7.1.3 Controlling Proximity

Mastering the usage of proximity gives the ability to locate results with greater precision. The Webinator input form gives you several options to control the search proximity:

`line` - All query terms must occur on the same line

`sentence` - Query items should all reside within the same sentence

`paragraph` - Within the same paragraph or text block

`page` - All items must occur within same HTML document (the default)

Note that the **Proximity** options may not be present (i.e. default to `page`) if it is disabled by the search administrator.

7.1.4 Ranking Factors

The ranking algorithm takes into consideration relative word ordering, word proximity, database frequency, document frequency, and position in text. The relative importance of these factors in computing the quality of a hit can be altered under `RANKING FACTORS` on the `Options` page.

7.1.5 Keywords Phrases and Wild-cards

To locate words, just type them in as you would in a word processor. Letter cases will be ignored.

The wild-card character $\star$ (asterisk) may be used to match just the prefix of a word or to ignore the middle of something.

If the item you wish to locate is more complicated than the simple $\star$ wild-card can accomplish, try using the regular expression matcher (http://www.thunderstone.com/texis/site/pages/regexp.html).

To locate a number of adjacent words in a specific order, surround them with `"` (double quotation) characters. Putting a − (hyphen) between words will also force order and one word proximity.

* see `Word Forms` (7.2, p. 221)

Table 7.1: Query examples

Query	Locates
`john`	john, John
`"john public"`	John Public
`web-browser`	Web browser, web-browser
`John*Public`	John Q. Public, John Public
`456*a*def`	1-456-789-ABCDEF
`activate`	activate, activation, activated, ... *

7.1.6 Applying Search Logic

Texis and Metamorph – the search software underlying Webinator – use set logic for text queries. The default behavior of the search is to locate an intersection (i.e. "AND") of every element within a query[1]. This means that the query: "`microsoft bob interface`" is the equivalent to the boolean query: "`microsoft AND bob AND interface`". The operators below modify this behavior:

- (without) The − (minus) is the most commonly used logic symbol[2]. It means the results *must exclude* those with that item.

+ (mandatory) The + (plus) symbol in front of a search item means that the results *must include* that item. This is generally used in conjunction with the intersection (`@`) operator.

@N (intersections) The `@` sign followed by a number[3] indicates how many intersections to locate of the other terms in the query (those without − or +). N intersections means that at least $N + 1$ distinct query terms must be present. This may be confusing at first, but it is powerful, as it enables arbitrary "partial" matches and combinations.

Table 7.2: Search Logic Examples

Query	Finds
`bob sam joe`	Bob with Sam and Joe
`bob sam -joe`	Bob with Sam without Joe
`bob sam joe @1`	Bob with Sam, or Bob with Joe, or Joe with Sam
`A B C D @1`	A B or A C or A D or B C or B D or C D
`A +B C D @0`	B and any of (A C or D)
`A B C -D @1`	(A B or A C or B C) without D

[1]For sort-by-relevance queries, this is true if **Require All Words** is `Y` in Search Settings, which is the default.
[2]It must be enabled via **Allow "NOT" Logic** in Search Settings.
[3]This must be enabled via **Allow the @ Operator** in Search Settings.

The plus (+) and minus (−) operators must immediately prefix the term to which they apply. There must be a space between the operator and any preceding term.

Correct	Incorrect
`bob +sam -joe`	`bob + sam - joe` `bob+sam-joe`

Note that instead of a single keyword, each term above could also be an entire set of things, or any of the special pattern matchers (e.g. REX). Such a set is present (as a term for search logic purposes) if any of its listed items match – just as a plain keyword is present if any of its suffix forms match (depending on **Word Forms**, p. 221)).

Specific lists of words are given within parentheses, separated by commas (with no spaces). For example: "`(bob,joe,sam)`" would match any of those words (without suffix processing). Logic operators apply to the entire set; thus "`(bob,robert,bobby) sue (elizabeth,liz,beth) @1 +(red,green,blue)`" would require (+) any of "red", "green", or "blue", and any two (`@1`) of Bob, Sue or Elizabeth – by any of their synonymous names.

7.1.7 Natural Language Query

You may enter a query in the form of a sentence or question. The software will automatically identify the important words and phrases within your query and remove the "noise words".

Example: What is the state of the art in text retrieval?

The software will search for: `state of the art` AND `text` AND `retrieval`

7.1.8 Using the Special Pattern Matchers

These pattern matchers are used to locate hard-to-find items within text:

- Regular expression matching for complex patterns
  `http://www.thunderstone.com/texis/site/pages/regexp.html`
- Approximate pattern matching for fuzzy searches
  `http://www.thunderstone.com/texis/site/pages/xpm.html`
- Numeric pattern matching for finding quantities
  `http://www.thunderstone.com/texis/site/pages/npm.html`

If improperly used these pattern matchers can slow queries. Therefore they require other keyword(s) in the query and are disabled entirely under Page proximity. For more details see the Vortex manual on Query Protection (`http://docs.thunderstone.com/site/vortexman/link_qprot.html`).

Table 7.3: Pattern Matcher Examples

Query	Matcher	Finds
`ronald %regan`	Approx	Ronald Raygun, Ronald Re-an, Ronald 8eagan
`%75MYPARTNO9045d/6a`	Approx	Anything within 75% of looking like MYPARTNO9045d/6a
`/19[789][0-9]`	RegEXpr	1970-1999
`/[1-9]{3}\-=[0-9]{4}`	RegEXpr	Phone numbers: 555-1212, 820-2200
`#87`	Numeric	four score and seven, 87
`#>0<1`	Numeric	Fractions like 9/16, 55%, 0.123, 15 nanoseconds

7.1.9 Invoking Thesaurus Expansion

Webinator has a vocabulary of over 250,000 word and phrase associations. Each entry is generally classifiable by either its meaning or part of speech.

Depending on the administrator's Synonyms setting for this profile, synonyms may already be included for each term in your query. If not, synonyms may be included for individual terms within your query by preceding them with a ~ (tilde) character.

7.2 Using Word Forms

The `Word forms` options give you control over how many variations of your query terms will be sought in your search.

Exact match: Only exact matches will be allowed. (the default)

Plurals & possessives: Plural and possessive forms will be found. (s, es, 's)

Any word forms: As many word forms as can be derived will be located.

Custom: Uses the `Custom Suffix List`, `Custom Suffix Default Removal`, and `Custom Suffix Min Length` settings to create your own custom behavior.

We call this morpheme processing, and it is generally smarter than a traditional "stemming" algorithm. It doesn't just rip the end off a word, it actually checks to see if it could be a valid form of the search term. More information is available at
`http://docs.thunderstone.com/site/texisman/link_ling.html`.

Notes: Thesaurus terms are also treated in the same manner. Words smaller than 4-5 characters will not be morpheme processed.

7.3 Controlling Proximity

These options give you control over the region in which a match must be found.

Table 7.4: Word Form Examples

Word	president
EXACT	president
PLURAL	(above) + presidents president's
ANY	(above) + presidential presidency preside presides presiding presided
Word	tight
EXACT	tight
PLURAL	(above) + tights
ANY	(above) + tightly tightening tightened tighter tightest
Word	program
EXACT	program
PLURAL	(above) + programs program's
ANY	(above) + programming programmatic programmed programmer programmable

`line` - match terms must be located within the same line

`sentence` - all terms within the same sentence

`paragraph` - match terms must be located within the same paragraph

`page` (default) - all terms within the same document

Note that the **Proximity** options may not be present (i.e. default to `page`) if it is disabled by the search administrator.

7.4 Interpreting Search Results

Note: *The look and feel described here is for the standard search interface. The interface may have been customized by the web site administrator.*

When a query is submitted it will come back with another query form and up to 10 matching documents. If there are more than 10 results, a link at the top and bottom of the list will allow you to view the next 10 in sequence.

The input form at the top allows you to further tailor your query to home-in on the desired results, or to submit a completely new query without having to navigate back to the original input form.

Each result in the set will have a format similar to the following:

```
1: THE DOCUMENT TITLE (hyperlink to original)    84%*******___
   This is the document abstract. It consists    Size: 11K
   of the text around the first hit within the   Depth: 3
   matching document...                          Find Similar
   http://www.example.com/thepage.html           Match Info
                                                 Show Parents
```

The components of each result are:

- Result number
- Document title *(Clicking on this will take you to the original document)*
- Abstract *(The first few hundred characters of the document)*
- Match quality graph. `84%*******____` *(Only shown if relevance ranking was used)*
- Size *(How big is the original document)*
- Depth *(How many clicks from the top of the site)*
- Find Similar *(Find other documents similar to this one)*
- Match Info *(View the matches and other information about the document)*
- Show Parents *(List pages that link to this one)*

7.4.1 Viewing Match Info

The `Match Info` link will show you the context of your results within the matching document. Matching words will be shown as hyperlinks. Clicking on any match term will take you to the next matching term. A summary at the top of the in-context view shows information about the document, including the time it was last modified.

7.4.2 Finding Similar Documents

The `Find Similar` link will find documents that are similar to the corresponding result. It does this by reading the original document to ascertain its main subject matter, and then conducting a relevance ranked search for those subjects.

Result documents are ordered from best to worst match. The bar graph display will indicate the overall quality of the match.

Note: The document you click on may not be ranked as the best match. This is because other documents may contain more information about the overall subject matter than the original.

7.4.3 Showing Document Parents

Often it is difficult to navigate using a search engine because there is no *back-link* present on the matching document. The `Show Parents` link solves this.

This link will show other documents that contain hyperlinks to the one you click on. In other words, it is an automated back button.