Manual for Setting Up and Using mySpider

Table of Contents

Eight Easy Steps

1.) Create a database and a database table and its appropriate fields as explained in “**The Database**” section, immediately below.
2.) Install all the mySpider files as outlined in “**Where the Files Go**”.
3.) Configure manually the “path_info.pl” file. See “**The path_info.pl File**”.
4.) Configure the two configuration data files. See “**Using the Two Configuration Data Files**”.
5.) Run the spider engine. See “**How to Spider and Re-spider**”.
6.) Integrate the Search feature into your web site’s design. See “**Integrating the Search Engine into Your Web Site and its Design**”.
7.) See “**Security**” for making your **mySpider** installation more secure.
8.) Do a test search.

♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦

The Database

9.) Before running the Spider or Search engines, you must create a single table in a database that will store the contents of the spidered web pages. This table is also referred to as the search index, and will be used by the Search engine. Even if your web site already has a database, for security purposes, you may want to create a separate database for the search index. You can name the database and the table anything you like. You could name the table "pages", for example. The name and location of the database, the name of the database table, and the username and password to access the database must be entered in the configure spider data and the configure search data files. See the section below, “**Using the Two Admin Tools and the Two Configuration Data Files**”.

This "pages" table must have three fields with the following properties:

Field Name	Data Type	Length	NULL	Unique	Default
url	varchar	255	no	yes	none
title	varchar	255	yes	no	none
body	text	(leave blank)	yes	no	none

Please note that if your database resides on a Windows server, and you’re not using a MySQL database, you must set up a DSN for the database name.

♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦

Where the Files Go

File Name	Windows	Unix
mySpider_spider (executable and referred to below as the spider engine) mySpider_search (executable and referred to below as the search engine) (both preferably, but not necessarily, in the same directory)	A sub-directory in the Web root directory.	cgi-bin or a sub-directory in the cgi-bin.
configure_mySpider_spider_data.pl configure_mySpider_search_data.pl (both in the same directory)	A directory outside the Web root directory. The path to this directory must be manually entered in the path_info.pl file.	A directory outside the Web root directory or else inside the cgi-bin. The path to this directory must be manually entered in the path_info.pl file.
path_info.pl (If you like, see **The path_info.pl File** section below for details.)	Web root directory. And a copy in the directory of the file* making the system call to the search engine.	Same directory as Spider & Search exe's, and in the directory of the file* making the system or include call to the search engine which could be the web root directory.
myspider_style.css	Web root directory.	Web root directory
All ".dll" files	Same directory as the Spider and Search executables.	N/A
All ".so" files	N/A	Same directory as the Spider and Search executables in cgi-bin.
myspider_call_spider.pl	Same directory as the Spider and Search executables.	Same directory as the Spider and Search executables.
myspider_results.txt	In the same directory as the 2 configure **data** files.	In the same directory as the 2 configure **data** files.
myspider_view_results.pl	Same directory as the Spider and Search executables.	Same directory as the Spider and Search executables.

pdfttotext (executable)	Web root directory	Same directory as the Spider executable.
this.pdf this.txt (both in the same directory)	In the same directory as the 2 configure **data** files.	In the same directory as the 2 configure **data** files.
spidercookies.txt	In the same directory as the 2 configure **data** files.	In the same directory as the 2 configure **data** files.

* If such a file is being used. See notes below on “Integrating the Search Engine.”

NOTE: The directory in which the two configure data files, the “this.pdf” and all the “.txt” files are kept, must have read/write permissions, but not executable (set CHMOD permissions to 0766).

On Unix, for all executables and all Perl files, except “path_info.pl” and the two configure data files in the “data” directory mentioned immediately above, set CHMOD permissions to 0755.

♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦

Individual File Information

NOTE: If you change the names of any files, **do not use** a hyphen in the new file name unless you have read the “**Multiple Secure Users and Host Companies**” manual.

NOTE: See the **Security** section below for why it is a good idea to change some file names.

NOTE: "CRON" below refers to any system task scheduler program. The "**shell**" refers to the command line.

File Name: mySpider-spider executable file.
Where it goes: **Windows:** a sub-directory of any name in the Web root directory.
Unix: cgi-bin or a sub-directory of any name in the cgi-bin.
What it does: Spiders your web site and gets URLs, titles and contents of all web pages and PDF documents; stores this info in database you create.
Can I Change the File Name? Yes. And the sub-directory you put it in can have any name.

File Name: mySpider-search executable file.
Where it goes: **Windows:** same directory as mySpider-spider.
Unix: same directory as mySpider-spider.
What it does: Searches for info stored in the database by mySpider-spider.
Can I Change the File Name? Yes.

File Name: All ".dll" files.
Where it goes: **Windows:** same directory as mySpider-spider.
Unix: N/A
What it does: Perl libraries used by the spider and search executables.
Can I Change the File Name? No.

File Name: All ".so" files.
Where it goes: **Windows:** N/A
Unix: same directory as mySpider-spider.
What it does: Perl libraries used by the spider and search executables.
Can I Change the File Name? No.

File Name: configure_mySpider_spider_data.pl
Where it goes: **Windows:** A directory outside the Web root directory.
Unix: A directory outside the Web root directory.
What it does: Configuration info used by the spider executable.
Can I Change the File Name? No. But you can give any name to the directory it's in.

File Name: configure_mySpider_search_data.pl
Where it goes: **Windows:** Same directory as configure_mySpider_spider_data.pl.
Unix: Same directory as configure_mySpider_spider_data.pl.
What it does: Configuration info used by the search executable.
Can I Change the File Name? No. But you can give any name to the directory it's in.

File Name: admin_mySpider_spider executable
or adminSSL_mySpider_spider executable
Best to use the SSL version only on a secure server.
Where it goes: **Windows:** same directory as the Spider and Search executables.
Unix: same directory as the Spider and Search executables**.**
What it does: Configures the data in configure_mySpider_spider_data.pl.
Can I Change the File Name? Yes.

File Name: admin_mySpider_search executable
or adminSSL_mySpider_search executable
Best to use the SSL version only on a secure server.
Where it goes: **Windows:** same directory as the Spider and Search executables.
Unix: same directory as the Spider and Search executables**.**
What it does: Configures the data in configure_mySpider_search_data.pl.
Can I Change the File Name? Yes.

File Name: path_info.pl
Where it goes: **Windows:** Web root directory and in the directory of the file making the system call. (See notes below on "Implementing the Search Engine.")
Unix: Same directory as the Spider and Search executables in the cgi-bin and in directory of the file making the system or include call. (See notes below on "Implementing the Search Engine.")
What it does: Contains the path to the configure data files used by the Spider and Search executables
Can I Change the File Name? No.

File Name: myspider_style.css
Where it goes: **Windows:** Web root directory.
Unix: Web root directory.
What it does: Style sheet for the appearance of the Search Engine.
Can I Change the File Name? No.

File Name: myspider_call_spider.pl
Where it goes: **Windows:** Same directory as the Spider and Search executables.
Unix: Same directory as the Spider and Search executables.
What it does: Used by a CRON or the shell, to execute the spider executable by making a GET request to it.(Note that you cannot execute the spider directly from the shell.)
Although you could run it in a browser, there would be no point.
Can I Change the File Name? Yes.

File Name: myspider_results.txt
Where it goes: **Windows:** In the same directory, outside the Web root directory, as the 2 configure **data** files.
Unix: In the same directory, outside the Web root directory, as the 2 configure **data** files.
What it does: Used by spider to store the progress and results of spidering.
Can I Change the File Name? No.

File Name: myspider_view_results.pl
Where it goes: **Windows:** Same directory as the Spider and Search executables.
Unix: Same directory as the Spider and Search executables in the cgi-bin.
What it does: Used to view the progress and results of spidering.
Can I Change the File Name? Yes.

File Name: pdfttotext executable
Where it goes: **Windows:** Web root directory.
Unix: Same directory as the Spider executable.
What it does: Used by Spider to extract text from PDF files.
Can I Change the File Name? No.

File Name: this.pdf **and** this.txt

Where it goes: **Windows:** In the same directory, outside the Web root directory, as the 2 configure **data** files.
Unix: In the same directory, outside the Web root directory, as the 2 configure **data** files.

What it does: Used by Spider to extract text from PDF files.

Can I Change the File Name? No.

File Name: spidercookies.txt

Where it goes: **Windows:** In the same directory, outside the Web root directory, as the 2 configure **data** files.
Unix: In the same directory, outside the Web root directory, as the 2 configure **data** files.

What it does: Used by Spider to store and read files.

Can I Change the File Name? No.

♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦

Additional Important Notes

1.) On Windows, be sure to keep an “index.html” or “default.html” file in the directory where the Search and Spider executables are stored, so no visitor to your web site will be able to access a listing of the directory’s files.

2.) In the “path_info.pl” file you must manually enter the full path to the directory where the two configuration data files are kept. Do not include the file names in this path and do not include a final forward slash after the name of the last directory in the path.

3.) For both Windows and Unix, the "this.pdf" and "this.txt" files, which are needed only if you are spidering PDF files, must be stored in the same folder as the two configure data files. This folder must have read/write permission. These two files must have read/write permission and due to security restrictions they cannot have executable permission.

♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦

Using the Two Configuration Data Files

The Spider and Search executables use the information stored in the "configure_mySpider_spider_data.pl" and the "configure_mySpider_search_data.pl" files respectively. If you have local access to the server they are on, you can edit the information stored in these files with a text editor. If they are on a remote server, you can:

1.) use SSH
2.) or download the files, edit them and upload them via FTP
3.) or edit them directly if your FTP client allows you to.

Both files contain descriptions and examples of the different entries and options.

The values to be set in the configuration_mySpider_search_data.pl:

database name
database location
table name
search user name
search password
maximum number of results per search page to be returned by the search
action (path and name to the script that calls the search exe, or "self")
maximum number of chunks of text or body excerpts returned
orig_maxchunklen (maximum number of characters in each chunk)
target window or frame that the linked titles of the search results will open in
categories or site sections visitors can select, one or more, to limit their search to
title-multiplier for search ranking
URL-multiplier for search ranking
body-multiplier for search ranking

The main values to be set in the configuration_mySpider_spider_data.pl:

database name
table name
database location
URL, the entry point for spidering the site
domain of the site that the spider limits its search to
authentication information for spidering protected areas
form login information for spidering protected areas
spider user name
spider password
file extensions you do not want spidered

names of files you do not want spidered

Notes re the configuration spider data file:

1.) It's best to enter an entry-URL for each domain to be spidered even though mySpider can use other domain URLs as entry-URLs.
2.) Always try to use the full URL to the entry pages, not just the URL to the directory, especially if the URL has an "html" or "htm" file extension. For example, use "http://www.you.com/secure/index.html" and not "http://www.you.com/secure/".
3.) In the "Form login information" enter all the field/value pairs necessary to login. The username and password fields may not be always be enough. Check your forms' hidden input elements.
4.) If a login uses an HTML form, enter that form's URL, not the processing page's URL. The spider will find the latter.
5.) If your web site has several login pages using one processing script, you will need customized code in the spider executable for the spider to "login" for each login.
6.) If any of your web site's pages contain redirects that are done with Javascript, the spider engine will need customized code to follow them.
7.) If your web forms redirect a visitor who has successfully logged-in, the page a visitor is redirected to after successfully logging-in should also include code that sends those not logged-in to the login page so they can log in.

♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦

Integrating the Search Engine into Your Web Site and its Design

You can add the Search function to your web site in five ways.

1.) In a frame set. The Search executable is set as a frame source in a frame set.
2.) In an iframe. The Search executable is set as the frame source for the iframe.
3.) The Search executable is linked directly from another web page.
4.) Call the Search executable from within the code of a web page that can make a "system" call to the Search executable. Typically this would be a PHP or ASP page for example, or even a Perl script. If it's a Perl web page in the "cgi-bin"

directory, all the links to other pages and images must be either set as relative to the web root directory (i.e. using “/”) or must be otherwise recoded.

5.) Use a PHP include, or the “WinHttpRequest” object in Classic ASP or the “WebRequest” object in ASP.NET. All three are shown further below.

The last two methods, numbers 4 and 5, are the best to way to integrate the Search function into the look and design of your web site.

If you are using a system command, PHP include, “WinHttpRequest” or “WebRequest” object to call the search engine executable from within another script or web page, DO NOT use a hyphen in that web page’s file name! Doing so, as in “search-page.php”, will cause the system call to fail, where as “search_page.php” will work. This is because in the search page’s file name is where the search engine executable looks for a hyphenated “client name”. See the “Secure Users Web Hosting Companies” Manual for an explanation of “client name” and when and how you can add a hyphenated string or name to the search page that is calling the search engine executable.

However, when using a system command, PHP include, “WinHttpRequest” or “WebRequest” object to call the search engine executable from within another web page, you must enter the name of that web page in the Admin search tool, in the section for the name of the script that is processing or calling the search executable. Enter that web page's file name including the absolute path from the web root, preceded by a forward slash, e.g. "/search.php" (without the quotes).

But if you are using a PHP include statement, “WinHttpRequest” or “WebRequest” object to a remote search engine executable residing in another domain, enter the absolute URL of the web page, e.g. "http://www.search_engine_domain.com/search-client1.php" (without the quotes).

If the search executable is NOT called by a system command, PHP include, “WinHttpRequest” or “WebRequest” object in another web page, or if it is used in an IFRAME or FRAME SET, enter "self" (without the quotes).

If you edit the configure search data file manually, the line to edit is the one for the $action variable.

In a PHP web page, the code for the system call to the search executable would be placed wherever the search results should appear in the web page, and could look like this:

```
<?php system("/var/path/to/the/cgi-bin/mySpider_search_myspider"; ?> (Unix)
<?php system("C:/path/to/the/mySpider_search_myspider.exe"); ?> (Windows)
```

In the above representative code snippet, “/path/to/the” represents the path’s directory structure to the search engine executable.

The code for the PHP include would look like this:

```
<?php
if (isset ($_SERVER['QUERY_STRING'])){$query_string = $_SERVER['QUERY_STRING'];}
else {$query_string="";}

if ($query_string != ""){include("http:// ww.hostcompany.com/pathto/mySpider_search.exe?$query_string");}
else  {include("http://ww.hostcompany.com/pathto/mySpider_search.exe ");}
?>
```

In Classic ASP, use the “WinHttp.WinHttpRequest” object as follows:

```
<%
dim strResult, strURL, temp
strURL="http:// ww.hostcompany.com /cgi-bin/mySpider_search.exe" & "?" & Request.QueryString
'Create the WinHTTPRequest ActiveX Object.
set WinHttpReq = CreateObject("WinHttp.WinHttpRequest.5.1")
'  Create an HTTP request.
temp = WinHttpReq.Open("GET", strURL, false)
'  Send the HTTP request.
WinHttpReq.Send()
'  Retrieve the response text.
strResult = WinHttpReq.ResponseText
'  Return the response text.
Response.Write (strResult)
Set winHttpReq = Nothing
%>
```

In ASP.NET, use the “WebRequest” object as follows:

```
<SCRIPT Language="VB" Option="Explicit" runat="server">
Dim strURI As String = "http:// ww.hostcompany.com / pathto/mySpider_search-client1.exe" & "?" & Request.QueryString
Dim objURI As URI = New URI(strURI)
Dim objWebRequest As WebRequest = WebRequest.Create(objURI)
Dim objWebResponse As WebResponse = objWebRequest.GetResponse()
Dim objStream As Stream = objWebResponse.GetResponseStream()
Dim objStreamReader As StreamReader = New StreamReader(objStream)
Dim strHTML As String = objStreamReader.ReadToEnd
Response.Write (strHTML)
</SCRIPT>
```

♦ ♦

The path_info.pl File

The tables below show the directories in which the path_info.pl and copies of it must be placed for use by the spider and search executables.

But note that because the spider and the search executables may look for the path_info.pl file in different directories, it is best to place copies of it in all the possible directories.

WINDOWS Platform
For the spider engine

Web root directory	Same directory as the spider engine
✓ Needed here when using a web browser to spider or when using a CRON or the shell to execute the myspider_call_spider.pl script that makes a GET request to run the spider. (Note that you cannot execute the spider directly from the shell.)	✗

WINDOWS Platform
For the search engine

Web root directory	Same directory as the file making a system call to the search engine
✓ Needed here if the search engine is not called by another script.	✓ Needed here when using a web page (PHP or ASP) that makes a system call to the search engine. If that web page is in the web root directory, then path_info.pl must be in the web root directory.

You can see from the above that for the Windows platform, the path_info.pl file must always reside in the web root directory.

UNIX Platform
For the spider engine

Web root directory	Same directory as the spider engine
	✓ Needed here when using a web browser to spider or when using a CRON or the shell to execute the myspider_myspider_call_spider.pl script that makes a GET request to run the spider. (Note that you cannot execute the spider directly from the shell.)

UNIX Platform
For the search engine

Web root directory	Same directory as the search engine	Same directory as the file making a system call to the search engine
✓ Needed here when using a web page (PHP or ASP) that makes a system call to the search engine, if that web page is in the web root directory.	✓ Needed here when the search engine is being run directly and not from a system call in a web page.	✓ Needed here when using a web page (PHP or ASP) that makes a system call to the search engine. If that web page is in the web root directory, then path_info.pl must be in the web root directory.

♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦

How to Spider and Re-spider

You can run the spider engine in any of three ways:

1. Run the spider executable in a web browser.
2. Run the myspider_call_spider.pl file in the shell.
3. Run the myspider_call_spider.pl file using a system task scheduler.

The first time you run the spider you should run it in a web browser to be sure the spider runs error free or to note any error messages.

1. **Run the spider executable in a web browser:**

Just point your browser to the spider executable. While the spider executable is running, do not refresh the browser window.

To view the spidering progress and any possible error messages, open a second browser window and open the script that whose file name was originally, and may still be, “myspider_view_results.pl”. You can and should refresh at your discretion the browser window that this file is running in so that you can see the spidering progress and results.

2. **Run the myspider_myspider_call_spider.pl file in the shell.**

You cannot run the spider executable from the command line. Instead, you can run “myspider_call_spider.pl” from the command line which will run the spider executable.

However, you must first open the source code of “myspider_call_spider.pl” and enter the absolute URL of the spider executable, and any username/password info if required to access the spider executable.

To view the spidering progress and any possible error messages, open a browser window and open the script that whose file name was originally, and may still be, “myspider_view_results.pl”. You can and should refresh at your discretion the browser window that this file is running in so that you can see the spidering progress and results.

3. **Run the myspider_call_spider.pl file using a system task scheduler.**

You cannot run the spider executable from the command line. Instead, you can use a system task scheduler program to run “myspider_call_spider.pl” which will in turn run the spider executable.

However, you must first open the source code of “myspider_call_spider.pl” and enter the absolute URL of the spider executable, and any username/password info if required to access the spider executable.

No matter which of the above methods you use, every time you run the spider, you may want to save the results as configured in the configure spider data file. Information such as the start, end date/times and time it took to spider, the list of bad links or of links outside the spidered domains is stored in the “myspider_results.txt” file. As noted above, you can view these results by running “myspider_view_results.pl” in a web browser and save the contents as an HTML file in a log directory or add the contents to a log file you have created.

You could also schedule the running of an email script after the scheduled spider run that reads the contents of the “myspider_results.txt” file and emails the results to a designated recipient. This script could also automatically save the results to a log file or directory.

Seamless Spidering

Visitors to your web site can use the search engine without interruption while the spider engine re-spiders your site.

After gathering all the links from a valid URL, the spider engine deletes the URL’s record, if it exists, from the search index. It then creates a new record, inserting the URL, title and body contents. Of course, if no such URL exists in the search index, then the insert occurs without any deletion. In either case, visitor’s can always do virtually a full search of your web site.

♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦

Security

There are a few security measures you should take that will prevent malevolent outsiders from accessing your mySpider executables, other files and sensitive data.

1. Change the name of certain files.
   (see the **Individual File Information** section above to review where each file resides and what it does.)

   - mySpider_spider
   - mySpider_search
   - myspider_call_spider.pl
   - admin_mySpider_spider.pl
   - adminSSL_mySpider_spider.pl
   - admin_mySpider_search.pl
   - adminSSL_mySpider_search.pl
   - myspider_view_results.pl

The new name must **not** contain a hyphen unless you have first read the “**Multiple Secure Users and Host Companies**” Manual and know which file names to alter.

Otherwise the new name can use any legal file name character. For example, the new “mySpider_spider” executable name could be “myspider_spider123_aBc”.

2. If your web site already is making use of one or more databases, use a separate database for the mySpider search index.

3. You could even keep the spider and search executables in different directories, and make the spider’s directory password protected with basic authentication. Of course each directory would also contain the “.so” or “dll” files and for Unix a copy of “path_info.pl”.

4. You could also choose to store the spider executable and the “.so” or .dll” files, “path_info.pl” and the configuration spider data file on your local hardrive instead of on the web server and spider from there. You would, however, need to set permissions in the MySql database for your hard drive to access the database remotely.

♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦

Altering the Look of the Search Page

Open the “search_page_css_example.html” file in a web browser to see which areas of the Search results page are affected by the different HTML Selectors in the “myspider_style.css” file.

♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦