

**Manual for  
Multiple Secure Users  
and for  
Web Hosting Companies**

**Table of Contents**

Introduction .....2

Multiple Secure Users .....2

Web Host Companies.....3

Database for Each Client.....3

File Naming Convention for Each Set of Client Files .....3

Security .....7

## Introduction

We can do for a lot less money what Google does for a lot more money! And then some!

Google's top of the line and very, very expensive Search Appliance, a combined hardware and software solution, offers Secure Search. Visitors can search for secure information and view only those documents to which they have access and search results are returned securely for documents protected by either NTLM or basic HTTP authentication.

Because the spider engine can access directories and files protected by basic authorization or by login forms, mySpider can create separate search indexes, each of which would contain contents for general viewing and contents requiring differing security access permissions. For example, in search index #1, some URLs would be for all visitors to view and some only for visitors who have level #1 access. In search index #2, some URLs would be for all visitors to view and some only for visitors who have level #2 access. A different search engine would be used for each different search index.

Similarly, Hosting companies can provide spider and search engines to their clients.

**You must read the “myspider\_manual.pdf” before reading and implementing any of what is contained in this manual.**



## Multiple Secure Users

To create different search indexes for the different levels of security, you need to purchase only one copy of the mySpider set of files. For each level of security you make a copy of the spider and search engines and all the other files, except the four Administration files. All copies of “path\_info.pl” contain the same code as the one main version that you configure manually. You individually configure the spider and search engines for each level of security. Put each level of security's spider and search executables and their required files into different directories protected by basic authentication. The search executable that searches the web areas that are open to all visitors doesn't need this protection, although the spider executable should have some form of security protecting it.

**In all the instructions below, the term “client” refers to each of the different levels of security.**



## Web Host Companies

Hosting companies can provide a non-customized version of mySpider, or the same customized version, to their clients on different domains. As well, any of their clients' spider and/or search engines can be customized if necessary.

Each client's spider and search engine can be stored and run only on the Hosting Company's domain.

You need to purchase only one copy of the mySpider set of files. For each of your clients you make a copy of the spider and search engines and all the other files. All copies of "path\_info.pl" contain the same code as the one main version that you configure manually. Each of your clients can individually configure their own spider and search engines that would reside on your domain, which is the Host Company's domain.

**In all the instructions below, the term "client" refers to each different client of the Host Company that will have a search feature for their web site.**



### Database Tables for Each Client

First, a new database table for each client must be set up. The name of each client's database table must be entered in their spider and search configuration data files.



### File Naming Convention for Each Set of Client Files

Each "configure\_mySpider\_spider\_data.pl" is renamed according to the name of the client. The spider configuration data file for the client CNN, for example, could be "configure\_mySpider\_spider\_data-CNN.pl". You can rename the part after the hyphen anything you like. But you cannot alter the filename of either the configure spider data or the configure search data file before the hyphen and **there can be only one hyphen in the entire file name.**

The name that comes after the hyphen is the part of the file name which the spider and search engines look for and recognize.

How do the spider and search engines know to look for this particular file name? Because the spider and search executables are programmed to look for certain files that have the same hyphenated name as they do. In this case "myspider\_spider-CNN.exe" would look

for the files it needs ending with “-CNN”. Again as stated above, you can rename the part after the hyphen any way you like, but **there can be only one hyphen in the entire file name**.

Regarding the spider and search executables, for security purposes, you can alter the filename before the hyphen as well. Each particular spider engine uses the hyphenated part of its own name to determine which configuration data file to use as well as which of the other files it requires.

If you use an Admin tool to make changes to a configuration data file, you will need to enter the client’s name as well as your login username and password in the Admin tool’s login window. And if you want to view or reconfigure more than one client’s configuration data files, you will have to open an Admin tool in a new web browser window for each client.

For the purpose of example only, the client name below is “client1”.

<b>Original File Name:</b>	<b>Changed to:</b>	<b>Or for greater security to:</b>
mypider_spider	mypider_spider-client1	mypider_spider_v1-client1
mypider_search	mypider_search-client1	mypider_search_b3-client1
mypider_view_results.pl	mypider_view_results-client1.pl	mypider_view_results_K8j-client1.pl

Below is a list of the files whose names you **MUST** change when you add a hyphenated client name to the names of the spider and search executables and to “mypider\_view\_results.pl”.

Note that the only change you can make to the names of the files below is to add the hyphenated client name.

<b>Original File Name:</b>	<b>MUST be changed to:</b>	<b>Or for greater security to:</b>
configure_mySpider_spider_data.pl	configure_mySpider_spider_data-client1.pl	✗
configure_mySpider_search_data.pl	configure_mySpider_search_data -client1.pl	✗
mypider_results.txt	mypider_results-client1.txt	✗
this.pdf	this-client1.pdf	✗
this.txt	this-client1.txt	✗
spidercookies.txt	Spidercookies-client1.txt	✗
mypider_style.css	mypider_style-client1.css	✗

The “.so” or “.dll” and the “path\_info.pl” file names must NOT change. The hyphenated client name is NOT added on to the end of these files.

If you are making a **system** call to the search engine executable from within another script or web page, the hyphenated client name must be added to the name of that web page, because in that web page’s file name is where the search engine looks for the client name. For example, if “search.php” is calling “myspider\_search-client1.exe”, then “search.php” must be renamed “search-client1.php”.

Do NOT add a hyphenated client name, or any hyphenated name or string, to “search.php” if you are not adding a hyphenated client name to the search engine executable! Doing so, as in “search-page.php”, will cause the system call to fail, where as “search\_page.php” will work.

The file named “search-client1.php” would contain:

```
<?php system("/var/path/to/the/cgi-bin/mySpider_search_myspider-client1"; ?> (Unix)
<?php system("C:/path/to/the/mySpider_search_myspider-client1.exe"); ?> (Windows)
```

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

If the client prefers not to use a system command to integrate the search tool into the design of their web site or they cannot use a system command from within a script in their domain to execute the mySpider search executable residing in the Hosting Company’s domain, they can instead use a PHP include statement to the absolute URL of the search executable. In this case, you must capture the query string and add it to the URL in the PHP include. 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 ($query_string != ""){
    include("http:// ww.hostcompany.com/pathto/mySpider_search-client1.exe?$query_string");
}
else {include("http://ww.hostcompany.com/pathto/mySpider_search-client1.exe ");}

?>
```

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

```
<%
dim strResult, strURL, temp
strURL="http:// ww.hostcompany.com / pathto/mySpider_search-client1.exe" & "?" & Request.QueryString
```

```
'Create the WinHttpRequest ActiveX Object.
set WinHttpRequest = CreateObject("WinHttp.WinHttpRequest.5.1")
```

```
' Create an HTTP request.
```

```
temp = WinHttpRequest.Open("GET", strURL, false)

' Send the HTTP request.
WinHttpRequest.Send()

' Retrieve the response text.
strResult = WinHttpRequest.ResponseText

' Return the response text.
Response.Write (strResult)
Set winHttpRequest = 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 / path/to/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>
```

But remember, when you use 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-client1.php” (without the quotes).

However, 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.



## Security

**Hosting companies** don't want "outsiders" or even a client to be able to run another client's spider engine by correctly guessing the name and location of the spider executable.

- 1.) Give each client's spider engine and "myspider\_view\_results.pl" unique file names before the hyphenated client name, as indicated in the first table above.
- 2.) Store each client's spider executable (and all the files it requires) in a different directory and protect that directory with basic authentication. Each client's search executable (and all the files it requires) is stored in a different and unprotected directory.
- 3.) Use both methods above.

**For different levels of security** on a single web site, you don't want anyone except yourself to be able to run any of the spider engines by correctly guessing the name and location of the spider executable. This is especially possible when the search executable's name and location may be visible to visitors when running the search tool for their level of security.

- 1.) Give each security level's spider engine and "myspider\_view\_results.pl" unique file names before the hyphenated client name, as indicated in the first table above.
- 2.) Store each security level's spider executable (and all the files it requires) in a different protected directory than the directory its search executable is stored in.
- 3.) Use both methods above.

The Admin files, for the sake of security, should be stored together with the above files. But if they are stored in the root cgi-bin, the file names before the hyphenated part should be changed and each client should easily recognize the file name of their own Admin files.

All the "configure\_mySpider\_spider\_data.pl" files for all clients must be stored in the same data directory of any name, located outside of the root web directory. The name and path of this directory is entered manually in the "path\_info.pl" file.

