STARDEVELOPER
.COM

7/12/2001

Quick Links : home | resources | book store | forum | whats new | advertise | contact us

PROFESSIONAL
Active Server Pages 3.0

Sample Chapter

News Letter

www.
wantjava
.com

FREE WEB
RESOURCES
CLICK HERE

ASP Search Engine

Author : Faisal Khan.

Please use a Java Script enabled browser.

In this article I will guide you through the steps of creating a simple but effective search engine. We will divide our work into two steps :

- **Step 1 :** Create a set of ASP pages to index the site content.
- **Step 2 :** Create a search engine to offer keyword specific database dependant search to our visitors.

Requirements :

- PWS / IIS
- Microsoft Access Database

We will begin with creating a set of ASP pages to index the site content and then insert into the database. Here you will learn how easy it practically is to deal with databases in ASP. All statements of SQL select, insert, update and delete will come into play.

Creating HTML Form Page :

We begin by creating an HTML Form page called 'addtodb.htm'. As it's name suggests you will use it to enter URLs of pages to index. So create a new HTML page and name it 'addtodb.htm' and then copy paste the following HTML code into it and then save it :

```
<html>
<head>
<style>
  body { font-family : Verdana; font-size : 8pt; }
  input { font-family : Verdana; font-size : 8pt; height : 20; width : 250; }
</style>
</head>

<body>
Insert / Update File : ( using HTTP )
<form action="addtodb.asp">
<input type="text" name="look_for"><br>
<input type="submit" value=" " style="height : 17; width : 17;"> Submit ->
</form>

<br><br><br>
Insert / Update File : ( using FileSystemObject )
<form action="addtodb_fso.asp">
Base URL :-<br>
<input type="text" name="base_url"><br>
Absolute path to the File :-<br>
<input type="text" name="look_for"><br>
<input type="submit" value=" " style="height : 17; width : 17;"> Submit ->
```

```
    </form>

    <br><br><br>
    Delete File :
    <form action="delfromdb.asp">
    <input type="text" name="del"><br>
    <input type="submit" value=" " style="height : 17; width : 17;"> Submit ->
    </form>
    </body>
    </html>
```

What does Form 1 do ?

As you might have guessed by seeing the code, it offer three Forms to site admin (you). The first one is used to enter complete URLs including HTTP (e.g. http://www.stardeveloper.com/default.asp). This Form will take you to 'addtodb.asp' page which will index the page using HTTP protocol. This is the easiest and most effective way to index pages. It enables you to index any of your site pages spanning across multiple hosts. It is also useful when title, description and keyword tags info in your pages is dynamically generated.

What does Form 2 do ?

The second Form instead of asking the complete URL (like the first Form), asks for base URL and absolute path to the page to index. This Form is optional. You should use it only when you cannot use HTTP protocol to index site pages (more on that). In the base URL field, you should enter the base URL of your site e.g. http://www.yoursite.com. And in the absolute path field, you should enter the absolute path to the page on your site e.g. /default.asp. This Form will lead to 'addtodb_fso.asp' page which indexes the pages using FileSystemObject. FileSystemObject is one of the scripting objects provided to you by ASP.
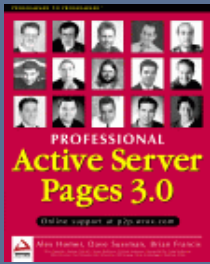
What does Form 3 do ?

The third Form asks for complete URL to the page e.g. http://www.stardeveloper.com/default.asp. You should use it when you want to delete a page entry in the database. Note this action will not delete the page, instead it will delete the indexed info of that page in the database. After this action that page will not be shown in the search engine.

In the next page we will create a simple Access database to use with our Search Engine. If you have read the database tutorial, it shouldn't be a problem.

Next ->

STARDEVELOPER
.COM

Quick Links : [home](#) | [resources](#) | [book store](#) | [forum](#) | [whats new](#) | [advertise](#) | [contact us](#)

Navigation

PROFESSIONAL
Active Server Pages 3.0

[Sample Chapter](#)

News Letter

www.
wantjava
.com

FREE WEB
RESOURCES
CLICK HERE

Simple Access Database

Creating Access Database :

Start Microsoft Access and create a new database 'directory.mdb'. Now create a table in the design view and name it 'all_pages'. Now put six fields in this table with 'id' field being the primary key. The names of the fields and their data types are shown in the image below :

all_pages : Table	
Field Name	Data Type
id	AutoNumber
title	Text
description	Memo
keywords	Memo
url	Text
mydate	Date/Time

Note if you are unsure or find it difficult, you can download the Access database discussed in this tutorial at the end. So stay cool.
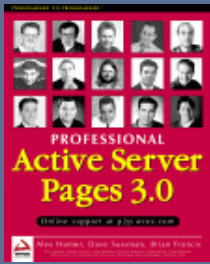
In the next page we will create 'addtodb.asp' page. It is the most important page of our search engine and will index the pages for us across all the web using HTTP protocol.

Please use a Java Script enabled browser.

[Next ->](#)

STARDEVELOPER
.COM

Quick Links :   [home](#) | [resources](#) | [book store](#) | [forum](#) | [whats new](#) | [advertise](#) | [contact us](#)

Navigation

PROFESSIONAL
Active Server Pages 3.0

[Sample Chapter](#)

News Letter

www.wantjava.com

FREE WEB RESOURCES
CLICK HERE

Indexing Site Pages Using HTTP Protocol

Creating 'addtodb.asp' page :

Please use a Java Script enabled browser.

Open note pad or your favorite text editor and create a new page. Save it as 'addtodb.asp'. Now copy paste the following code into it :

```
<!--#include file="editme.asp"-->
<html>
<head>
  <style>
    body { font-family : Verdana; font-size : 8pt; }
  </style>
</head>
<body>

<%   On Error Resume Next
  Dim geturl, title, description, keywords, strURL, strDB, con, results

    ' URL
    strURL = Request.QueryString("look_for")

    Set geturl = CreateObject("Stardeveloper.GetURL")
    strFileContents = geturl.Get(strURL)
    Set geturl = Nothing

    ' Keywords
    key1 = InStr(1, strFileContents, "<meta name=""keywords"" content=""", 1)
    key1 = key1 + Len("<meta name=""keywords"" content=""")
    key2 = InStr(key1, strFileContents, """>", 1)

    keywords = "," & Trim(Mid(strFileContents, key1, (key2 - key1))) & ","
    keywords = Replace (keywords, """", " ")

    ' Description
    desc1 = InStr(1, strFileContents, "<meta name=""description"" content=""", 1)
    desc1 = desc1 + Len("<meta name=""description"" content=""")
    desc2 = InStr(desc1, strFileContents, """>", 1)

    description = Trim(Mid(strFileContents, desc1, (desc2 - desc1)))
    description = Replace (description, """", " ")

    ' Title
    tit1 = InStr(1, lcase(strFileContents), "<title>", 1)
    tit1 = tit1 + Len("<title>")
    tit2 = InStr(tit1, strFileContents, "</title>", 1)
```

```
        title = Trim(Mid(strFileContents, tit1, (tit2 - tit1)))
        title = Replace (title, """, " ")


        ' Our Connection Object
        Set con = CreateObject("ADODB.Connection")
        con.Open strDB
        Set results = con.Execute("select title, description, keywords _
                from all_pages where url = '" & strURL & "'")


     ' If the returning recordset is empty the add the URL with accompanying
     ' info to the database
     If results.EOF Then
        con.Execute("insert into all_pages (title, description, keywords, url, _
                mydate) values ('" & title & "', '" & description & "', '" & _
                keywords & "', '" & strURL & "', '" & date & "')")
        Set rs = con.Execute("select count(url) as total_count from all_pages")
        cnt = rs("total_count")
        Set rs = Nothing
        Response.Write "<b>New account successfully created for " & strURL & " _
                .</b>" & vbcrlf
        Response.Write "<br>"
        Response.Write "Total Pages Indexed : " & cnt & vbcrlf
     Else
     ' But if the returning recordset is not empty i.e. we have already added _
     ' title, desc, keywords etc into it then update that information with the _
     ' new one.
        con.Execute("update all_pages set title = '" & title & "', description = _
            '" & description & "', keywords = '" & keywords & "', mydate = #" _
            & date & "# where url = '" & strURL & "'")
        Response.Write "<b>Account updated successfully.</b>"
     End If


     ' Done. Now release Objects
     Set results = Nothing
     con.Close
     Set con = Nothing

%>
  </body></html>
```

Reviewing the code :

We will now very quickly look into the code and see how it's done. Although this page might seem a bit long, but if you carefully look into it, all it contains are simple VBScript function calls and nothing more than that.

```
  <!--#include file="editme.asp"-->
```

This is first line of the page. All it does is to include 'editme.asp' file into this page. 'editme.asp' contains a single variable to store the relative location of the database. We will look into 'editme.asp' page later.
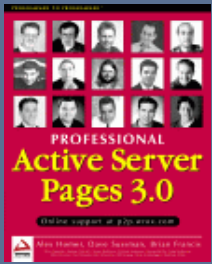
On the next page we continue with the exploration of 'addtodb.asp' page.

© 1999 - 2001 Stardeveloper.com.

STARDEVELOPER
.COM

Quick Links :   home | resources | book store | forum | whats new | advertise | contact us

PROFESSIONAL
**Active Server
Pages 3.0**

Sample Chapter

News Letter

www.
wantjava
.com

**FREE WEB
RESOURCES**
CLICK HERE

Indexing Site Pages Using HTTP Protocol

Reviewing the code :

Please use a Java Script enabled browser.

```
<html>
<head>
  <style>
    body { font-family : Verdana; font-size : 8pt; }
  </style>
</head>
<body>
```

Above few lines are HTML tags with no ASP content.

```
<%    On Error Resume Next
```

Tells ASP script interpreter to continue with the next statement and not to stop if it receives an error during script execution. Actually it consists of two parts :

```
On Error
```

By writing it we tell ASP script that we will receive the error and we will decide whether to stop the script from further execution (by doing nothing) or do some thing with it. On Error is an event which is fired when an error occurs in the execution of the script.

```
Resume Next
```

By writing this we decide that we want ASP script interpreter to continue with the execution of the script and don't stop if it receives an error. This discussion has got a bit long, but it is very important that you understand what this statement does since you will be seeing it quite often in the ASP scripts, so you should be at ease with it. Tip, when you are testing your ASP code knock this line off, this will let you see the error and make corrections, but when you are releasing your ASP code then write this statement on top of your ASP page so that your script users might not see the errors if produced.

```
Dim geturl, title, description, keywords, strURL, strDB, con, results
```

We define a list of variables which we will be using later in our script. Note in ASP all the variables are of type VARIANT, which means they can be of any type e.g. string (text), int (number), object etc.

```
' URL
strURL = Request.QueryString("look_for")
```

strURL receives the URL which will be entered in the Form 1 of 'addtodb.htm' page. Since Form 1 had only one field with name 'look_for', we receive it's value by using this field name in the Request.QueryString method.

```
Set geturl = CreateObject("Stardeveloper.GetURL")
strFileContents = geturl.Get(strURL)
Set geturl = Nothing
```

Now this is the real part to retrieve the contents of the page specified in the Form 1's 'look_for' field by

using HTTP method. Note that ASP uses server side scripting languages to generate dynamic content, it is not possible to retrieve the contents of any page on the web using scripting languages. Scripting languages simply don't have that power. So what do we do ? we use our own component to retrieve the content of that web page using HTTP protocol. This is where ASP is very handy. It lets us use COM components to do tasks which we cannot do with simple VBScript. So with this search engine I am including my own Free component which can by used to retrieve page contents into a variable using HTTP protocol. All you have to do to use it can be seen by looking at the above three lines. Yes that's right, it is very simple to use. Just create an instance of it and then use it's only method Get() to get the page contents in a variable. Note that the only argument to Get() is strURL variable which is the URL of the page we want to retrieve.

As you might be knowing that in order to use a COM component on your computer (or server), you have to register it using regsvr32 command. So in order to use this component go to the DOS prompt and then move to the location of the directory where you have kept the search engine files. Now type the following command :

```
regsvr32 GetURL.dll
```

And press enter. You will see a small window saying that component was successfully registered. That's it. Now you can use this component. Note, the ProgID of this component is Stardeveloper.GetURL.

```
' Keywords
key1 = InStr(1, strFileContents, "<meta name=""keywords"" content=""", 1)
key1 = key1 + Len("<meta name=""keywords"" content=""")
key2 = InStr(key1, strFileContents, """>", 1)

keywords = "," & Trim(Mid(strFileContents, key1, (key2 - key1))) & ","
keywords = Replace (keywords, """, " ")

' Description
desc1 = InStr(1, strFileContents, "<meta name=""description"" content=""", 1)
desc1 = desc1 + Len("<meta name=""description"" content=""")
desc2 = InStr(desc1, strFileContents, """>", 1)

description = Trim(Mid(strFileContents, desc1, (desc2 - desc1)))
description = Replace (description, """, " ")

' Title
tit1 = InStr(1, lcase(strFileContents), "<title>", 1)
tit1 = tit1 + Len("<title>")
tit2 = InStr(tit1, strFileContents, "</title>", 1)

title = Trim(Mid(strFileContents, tit1, (tit2 - tit1)))
title = Replace (title, """, " ")
```

Since above three paragraphs of ASP script are similar we will discuss them together. All they do is to check the availability of certain HTML meta tags within that page. If you are familiar with HTML then you must be knowing about title, description and keyword meta tags. Above code looks for the presence of these meta tags in the contents of that page (which is present in the strFileContents variable). Note that earlier we retrieved all the contents of that page in the strFileContents variable, so we use this variable when we look for meta tags.

Since we are looking for meta tags to index their contents, make sure that title, description and keywords tags are present in the <head></head> section of your page which you want to index in the following manner :

```
<title>Your Page Title Goes Here</title>
<meta name="description" content="Your page description goes here">
<meta name="keywords" content="keyword1, keyword2, keyword3, keyword4 ...">
```

So our above ASP script indexed the contents of given web page and then retrieved the values of title, description and keywords tags in to title, description and keywords variables respectively.
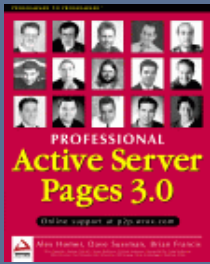
On the next page we continue with the exploration of 'addtodb.asp' page.

Next ->

STARDEVELOPER
.COM

Quick Links :   [home](#) | [resources](#) | [book store](#) | [forum](#) | [whats new](#) | [advertise](#) | [contact us](#)

Navigation

PROFESSIONAL
Active Server Pages 3.0

[Sample Chapter](#)

News Letter

www.wantjava.com

FREE WEB RESOURCES
CLICK HERE

Indexing Site Pages Using HTTP Protocol

Reviewing the code :

Please use a Java Script enabled browser.

```
' Our Connection Object
Set con = CreateObject("ADODB.Connection")
con.Open strDB
Set results = con.Execute("select title, description, keywords _
      from all_pages where url = '" & strURL & "'")
```

After populating the three variables title, description and keywords, we now move forward to add them to the database. We begin by creating Connection object. Then we open the database. Note as we discussed [earlier](#), 'editme.asp' contains the location of the database in the variable strDB. So after including 'editme.asp' page in our 'addtodb.asp' page we can easily use the value of strDB in our script.

We then check the database to see if this given URL has been added to the database before. We do it by selecting title, description and keywords fields from the 'all_pages' table where URL is the given URL. If the returning recordset is empty then it shows that the URL has not been added to the database. But if the returning recordset is not empty then it means that URL has been added to the database before.

```
' If the returning recordset is empty then add the URL with accompanying
' info to the database
If results.EOF Then
  con.Execute("insert into all_pages (title, description, keywords, url, _
      mydate) values ('" & title & "', '" & description & "', '" & _
      keywords & "', '" & strURL & "', '" & date & "')")
  Set rs = con.Execute("select count(url) as total_count from all_pages")
  cnt = rs("total_count")
  Set rs = Nothing
  Response.Write "<b>New account successfully created for " & strURL & " _
      .</b>" & vbcrlf
  Response.Write "<br>"
  Response.Write "Total Pages Indexed : " & cnt & vbcrlf
```

If the URL has not been added to the database before, we then add it along with the retrieved title, description and keywords values and current date.

```
Else
' But if the returning recordset is not empty i.e. we have already added _
' title, desc, keywords etc into it then update that information with the _
' new one.
  con.Execute("update all_pages set title = '" & title & "', description = _
      '" & description & "', keywords = '" & keywords & "', mydate = #" _
      & date & "# where url = '" & strURL & "'")
  Response.Write "<b>Account updated successfully.</b>"
End If
```

But if the URL has been added to the database earlier then instead of entering another row for this URL, update the earlier values in their respective fields. Note this is useful if you update a page and then also want to update it's info in the search engine database.

```
    ' Done. Now release Objects
    Set results = Nothing
    con.Close
    Set con = Nothing

%>
  </body></html>
```

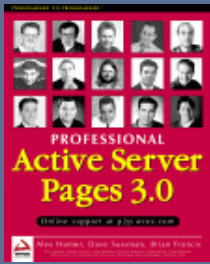Now we are done, so lets remove the objects from the server memory.

Summarizing, we retrieved the contents of given page in the variable strFileContents. Then we searched for the title, description and keywords meta tags and populated our title, description and keywords ASP variables with their values. In the end we added or updated their values in the database.

On the next page we explore 'addtodb_fso.asp' page which indexes the pages using FileSystemObject in a manner similar to 'addtodb.asp' page, but with a slight difference.

[Next ->](#)

STARDEVELOPER
.COM

7/12/2001

Quick Links :   home | resources | book store | forum | whats new | advertise | contact us

Sample Chapter

News Letter

Indexing Site Pages Using FileSystemObject

Creating 'addtodb_fso.asp' page

Create a new blank ASP page and name it 'addtodb_fso.asp'. Then copy paste the following code into it :

```
<!--#include file="editme.asp"-->
<html>
<head>
  <style>
    body { font-family : Verdana; font-size : 8pt; }
  </style>
</head>
<body>

<%   On Error Resume Next
  Dim geturl, title, description, keywords, strURL, strDB, con, results
  Dim fso, file, path

    ' URL
    If Trim(Request.QueryString("base_url")) <> "" And _
          Request.QueryString("look_for") <> "" Then
    strURL = Request.QueryString("base_url") & Request.QueryString("look_for")
    path = Server.MapPath(Request.QueryString("look_for"))

    ' FileSystemObject
    Set fso = CreateObject("Scripting.FileSystemObject")

    ' File Object
    Set file = fso.OpenTextFile(path, 1, False)

    ' Reading the contents of the File into strFileContents
    strFileContents = file.ReadAll

    ' Keywords
    key1 = InStr(1, strFileContents, "<meta name=""keywords"" content=""", 1)
    key1 = key1 + Len("<meta name=""keywords"" content=""")
    key2 = InStr(key1, strFileContents, """>", 1)

    keywords = "," & Trim(Mid(strFileContents, key1, (key2 - key1))) & ","
    keywords = Replace (keywords, """, " ")

    ' Description
    desc1 = InStr(1, strFileContents, "<meta name=""description"" content=""", 1)
    desc1 = desc1 + Len("<meta name=""description"" content=""")
    desc2 = InStr(desc1, strFileContents, """>", 1)
```

Please use a Java Script enabled browser.

```
            description = Trim(Mid(strFileContents, desc1, (desc2 - desc1)))
            description = Replace (description, """", " ")

            ' Title
            tit1 = InStr(1, lcase(strFileContents), "<title>", 1)
            tit1 = tit1 + Len("<title>")
            tit2 = InStr(tit1, strFileContents, "</title>", 1)

            title = Trim(Mid(strFileContents, tit1, (tit2 - tit1)))
            title = Replace (title, """", " ")

            ' Our Connection Object
            Set con = CreateObject("ADODB.Connection")
            con.Open strDB
            Set results = con.Execute("select title, description, keywords _
                    from all_pages where url = '" & strURL & "'")

            ' If the returning recordset is empty the add the URL with accompanying
            ' info to the database
            If results.EOF Then
              con.Execute("insert into all_pages (title, description, keywords, url, _
                    mydate) values ('" & title & "', '" & description & "', '" & _
                    keywords & "', '" & strURL & "', '" & date & "')")
              Set rs = con.Execute("select count(url) as total_count from all_pages")
              cnt = rs("total_count")
              Set rs = Nothing
              Response.Write "<b>New account successfully created for " & strURL & " _
                    .</b>" & vbcrlf
              Response.Write "<br>"
              Response.Write "Total Pages Indexed : " & cnt & vbcrlf
            Else
            ' But if the returning recordset is not empty i.e. we have already added _
            ' title, desc, keywords etc into it then update that information with the _
            ' new one.
              con.Execute("update all_pages set title = '" & title & "', description = _
                    '" & description & "', keywords = '" & keywords & "', mydate = #" _
                    & date & "# where url = '" & strURL & "'")
              Response.Write "<b>Account updated successfully.</b>"
            End If

            ' Done. Now release Objects
            Set results = Nothing
            con.Close
            Set con = Nothing

            Else
            Response.Write "Please enter the base URL and File path correctly. Thanks."
            End If
        %>
          </body></html>
```

Almost similar to 'addtodb.asp' with only difference being of the following code :

```
            ' URL
            If Trim(Request.QueryString("base_url")) <> "" And _
                    Request.QueryString("look_for") <> "" Then
            strURL = Request.QueryString("base_url") & Request.QueryString("look_for")
```

```
        path = Server.MapPath(Request.QueryString("look_for"))

        ' FileSystemObject
        Set fso = CreateObject("Scripting.FileSystemObject")

        ' File Object
        Set file = fso.OpenTextFile(path, 1, False)

        ' Reading the contents of the File into strFileContents
        strFileContents = file.ReadAll
```
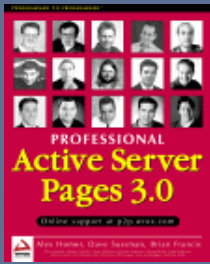
On the next page we continue with the exploration of 'addtodb_fso.asp' page.

Next ->

© 1999 - 2001 Stardeveloper.com.

STARDEVELOPER
.COM

7/12/2001

Quick Links : [home](#) | [resources](#) | [book store](#) | [forum](#) | [whats new](#) | [advertise](#) | [contact us](#)

PROFESSIONAL
Active Server Pages 3.0

[Sample Chapter](#)

News Letter

www.
wantjava
.com

FREE WEB
RESOURCES
CLICK HERE

Indexing Site Pages Using FileSystemObject

Reviewing the code :

Please use a Java Script enabled browser.

```
' URL
If Trim(Request.QueryString("base_url")) <> "" And _
     Request.QueryString("look_for") <> "" Then
```

Checks to see if the base URL and absolute path entered are empty or not. Note the base URL which you enter into 'base URL' field of Form 2 of 'addtodb.htm' page should be the base URL of your site you want to index e.g. http://www.yoursite.com or http://www.othersite.com/yoursitefolder etc.

And the absolute path should be the absolute path to the page on your site e.g. /default.asp or /folder/anotherpage.asp etc. So in FileSystemObject method of indexing site pages you have to enter two fields to make it work. Although only absolute path would have been enough to index a page but when we will be displaying the results to the user, there we will need complete path to the page so adding base URL is very important here.

```
strURL = Request.QueryString("base_url") & Request.QueryString("look_for")
```

We concatenate the given base URL and absolute path to the page to get strURL. It is this variable which will be inserted into the database later.

```
path = Server.MapPath(Request.QueryString("look_for"))
```

path contains the mapped physical location of the given page. Note Server.MapPath method translates the relative or absolute path of a file to a complete physical path to that file on that computer e.g. http://127.0.0.1/folder/file.asp will be translated to c:/inetpub/wwwroot/folder/file.asp.

```
' FileSystemObject
Set fso = CreateObject("Scripting.FileSystemObject")
```

Creates FileSystemObject.

```
' File Object
Set file = fso.OpenTextFile(path, 1, False)
```

Using FileSystemObject's OpenTextFile method we open the given page and create that page's file object.

```
' Reading the contents of the File into strFileContents
strFileContents = file.ReadAll
```

We then use File object's ReadAll method to read all the content of that page into the variable strFileContents variable. Rest of the code is the same as 'addtodb.asp' page.

On the next page we create 'delfromdb.asp' to delete specific URLs from the Search Engine database if you would later want to remove some pages from the database.
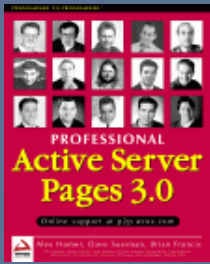
© 1999 - 2001 Stardeveloper.com.

STARDEVELOPER
.COM

Quick Links :   [home](#) | [resources](#) | [book store](#) | [forum](#) | [whats new](#) | [advertise](#) | [contact us](#)

Navigation

PROFESSIONAL
Active Server Pages 3.0

[Sample Chapter](#)

News Letter

www.
wantjava
.com

FREE WEB
RESOURCES
CLICK HERE

Deleting indexed URLs from the database

Creating 'delfromdb.asp' page :

Create a new ASP page and name it 'delfromdb.asp'. Copy and paste the following code into it :

```
<!--#include file="editme.asp"-->
<html>
<head>
  <style>
    body { font-family : Verdana; font-size : 8pt; }
  </style>
</head>
<body>

<%
Dim con, strDB

    ' URL
    strURL = Request.QueryString("del")

    ' Creating our Connection Object
    Set con = CreateObject("ADODB.Connection")
    con.Open strDB

    ' Now deleting the account
    con.Execute("delete * from all_pages where url = '" & strURL & "'")

    ' Telling the admin that account has been deleted
    Response.Write "<b>Account successfully deleted : " & strURL & " </b>."

    ' Done. Now release Objects
    con.Close
    Set con = Nothing

%>
    </body></html>
```

Reviewing the code :

```
<!--#include file="editme.asp"-->
<html>
<head>
  <style>
    body { font-family : Verdana; font-size : 8pt; }
  </style>
```

Please use a Java Script enabled browser.

```
      </head>
      <body>

 <%
   Dim con, strDB
```

Same as earlier pages.

```
    ' URL
    strURL = Request.QueryString("del")
```

strURL receives the URL entered in the Form 3 of 'addtodb.htm' page.

```
    ' Creating our Connection Object
    Set con = CreateObject("ADODB.Connection")
    con.Open strDB
```

Creates the Connection object and opens the database.

```
    ' Now deleting the account
    con.Execute("delete * from all_pages where url = '" & strURL & "'")
```

Deletes the record in the database of the given URL.

```
    ' Telling the admin that account has been deleted
    Response.Write "<b>Account successfully deleted : " & strURL & " </b>."

    ' Done. Now release Objects
    con.Close
    Set con = Nothing

 %>
      </body></html>
```

Tell the user that the given URL has been deleted. Then release the objects and end the page.
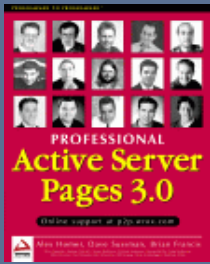
On the next page we create 'search.htm' to create a Form to show users to enter keywords to search pages from our Search Engine database.

[Next ->](#)

STARDEVELOPER
.COM

7/12/2001

Quick Links :   home | resources | book store | forum | whats new | advertise | contact us

PROFESSIONAL
Active Server
Pages 3.0

Sample Chapter

News Letter

www.
wantjava
.com

FREE WEB
RESOURCES
CLICK HERE

Search Engine Form to search records

Creating 'search.htm' page :

Please use a Java Script
enabled browser.

```
<html>
<head>
<style>
  body { font-family : Verdana; font-size : 8pt; }
  input { font-family : Verdana; font-size : 8pt; height : 20; width : 250; }
</style>
</head>

<body>

Search Records :
<form action="search.asp">
<input type="text" name="look_for"><br>
<input type="submit" value=" " style="height : 17; width : 17;"> Submit ->
</form>

</body>
</html>
```

Reviewing the code :

Simple HTML page with a single Form asking for the complete URL e.g.
http://www.yoursite.com/urpage.asp of the page to delete. Note the page will not be physically deleted
but only it's info in the database will be removed. You can copy paste the following code into every page
where you want to offer search engine Form to your users.

```
Search Records :
<form action="search.asp">
<input type="text" name="look_for"><br>
<input type="submit" value=" " style="height : 17; width : 17;"> Submit ->
</form>
```
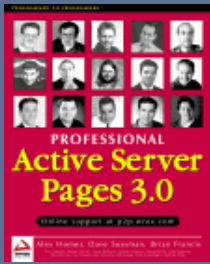
On the next page we create 'search.asp' to search the Search Engine database for the entered keywords.

Next ->

© 1999 - 2001 Stardeveloper.com.

**STARDEVELOPER
.COM**

Quick Links :   home | resources | book store | forum | whats new | advertise | contact us

Navigation

**PROFESSIONAL
Active Server
Pages 3.0**

Online support at p2p.wrox.com

Sample Chapter

News Letter

www.
wantjava
.com

**FREE WEB
RESOURCES**
CLICK HERE

Searching Records in the Search Engine Database

Creating 'search.asp' page :

Create a new ASP page and name it 'search.asp'. Copy paste the following code into it :

```asp
<!--#include file="editme.asp"-->
<!--#include file="adovbs.inc"-->
<html>
<head>
  <style>
    body { font-family : Verdana; font-size : 8pt; }
  </style>
</head>
<body>

<%
'On Error Resume Next
Dim geturl, title, description, keywords, strURL, strDB, rs, strKeyword
Dim urlshown, records, recordsToShow, n, con, currentPage, show

    If Len(Request.QueryString("currentPage")) = 0 Then
        currentPage = 1
    Else
        currentPage = Request.QueryString("currentPage")
    End If

    records = ""
    recordsToShow = 10
    n = 0

    ' Keyword to search
    strKeyword = split(Trim(Request.QueryString("look_for")), " ")

    ' Our Connection Object
    Set con = CreateObject("ADODB.Connection")
    con.Open strDB

    ' Our Recordset Object
    Set rs = CreateObject("ADODB.Recordset")
    rs.CursorLocation = adUseClient
    rs.CacheSize = recordsToShow
    show = True

    ' Searching the records for the keywords entered
    Select Case UBound(strKeyword)
```

Please use a Java Script enabled browser.

```
        Case 0 rs.Open "select * from all_pages where keywords like '%" _
            & strKeyword(0) & "%' order by mydate desc", con
        Case 1 rs.Open "select * from all_pages where keywords like '%" _
            & strKeyword(0) & "%' and keywords like '%" & strKeyword(1) & _
            "%' order by mydate desc", con
        Case 2 rs.Open "select * from all_pages where keywords like '%" _
            & strKeyword(0) & "%' and keywords like '%" & strKeyword(1) & _
            "%' and keywords like '%" & strKeyword(2) & "%' order by _
            mydate desc", con
        Case Else rs.Open "select * from all_pages where keywords like '%" _
            & strKeyword(0) & "%' and keywords like '%" & strKeyword(1) & _
            "%' and keywords like '%" & strKeyword(2) & "%' order by _
            mydate desc", con
        End Select

    ' If the returning recordset is not empty
    If Not rs.EOF Then

    records = records & "found"
    rs.PageSize = recordsToShow
    totalpages = rs.PageCount
    rs.AbsolutePage = currentPage

    ' Showing total number of pages found and the current page number
    Response.Write "Displaying Page " & currentPage & " of " & totalPages & "<br>"
    Response.Write "Total Records Found : " & rs.RecordCount
    Response.Write "<br><br>"

    ' Showing relevant records
    Do Until rs.EOF
    ' Showing only 10 records at a time
    If n = recordsToShow Then
    Exit Do
    End If

        If InStr(1, urlshown, rs("url"), 1) = 0 Then

        Response.Write "<b><a href=""" & rs("url") & """>" & rs("title") & _
            "</a></b><br>" & vbcrlf
        Response.Write "" & rs("description") & "<br>" & vbcrlf
        Response.Write "URL : " & rs("url") & "</b><br>" & vbcrlf
        Response.Write "Last indexed on : " & rs("mydate") & "<br><br>" & vbcrlf

        n = n + 1
        End If

        urlshown = urlshown & " " & rs("url") & " "
        rs.MoveNext
    Loop

    For i = 0 To UBound(strKeyword)
        strKeyword2 = strKeyword(i)
    Next
    look_for = strKeyword2

    ' Links to move through the records
    If currentPage > 1 Then
```

```
            Response.Write "<a href=""" & Request.ServerVariables("SCRIPT_NAME") & _
                "?currentPage=" & currentPage - 1 & "&look_for=" & look_for & _
                """>Back</a>"
        Else
            Response.Write "<u style=""color : silver;"">Back</u>"
    End If

    Response.Write "        "

    If CInt(currentPage) <> CInt(totalPages) Then
        Response.Write "<a href=""" & Request.ServerVariables("SCRIPT_NAME") & _
            "?currentPage=" & currentPage + 1 & "&look_for=" & look_for & _
            """>Next</a>"
        Else
            Response.Write "<u style=""color : silver;"">Next</u>"
    End If

    End If

    ' Done. Now release Objects
    con.Close
    Set con = Nothing
    Set rs = Nothing

  If InStr(1, records, "found", 1) = 0 Then
  Response.Write "<b>Sorry, no matching record was found.</b>" & vbcrlf
  End If
%>
  </body></html>
```
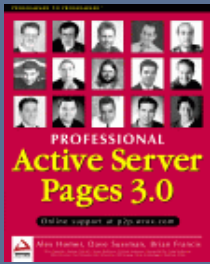
On the next page we will review the above code.

[Next ->](#)

© 1999 - 2001 Stardeveloper.com.

7/12/2001

Quick Links : home | resources | book store | forum | whats new | advertise | contact us

Searching Records in the Search Engine Database

Reviewing the code :

Please use a Java Script
enabled browser.

```
<!--#include file="editme.asp"-->
<!--#include file="adovbs.inc"-->
```

Includes two files into the page. You are already aware of 'editme.asp' and
'adovbs.inc' contains ADO constants.

```
<html>
<head>
  <style>
    body { font-family : Verdana; font-size : 8pt; }
  </style>
</head>
<body>
```

Simple HTML head and body tags.

```
<%
 On Error Resume Next
```

We discussed this statement earlier.

```
 Dim geturl, title, description, keywords, strURL, strDB, rs, strKeyword
 Dim urlshown, records, recordsToShow, n, con, currentPage, show
```

We declare a list of variables which we will be using later in our code.

```
 If Len(Request.QueryString("currentPage")) = 0 Then
     currentPage = 1
 Else
     currentPage = Request.QueryString("currentPage")
 End If
```

Looks for currentPage variable in the query string. If it doesn't find one (which means that it is the first
page) then it sets the currentPage variable to 1. Note currentPage variable is used when moving across
pages of records.

```
 records = ""
 recordsToShow = 10
 n = 0
```

We declare some variables here. Important is recordsToShow. You should change it to the number of
records you want to show in one page. By default it is set to 10. Which means that if more than 10
matching records are found in the Search Engine database, then it will display 10 of them in a page and
will give a 'Next' link to the next ten or so records.

```
 ' Keyword to search
 strKeyword = split(Trim(Request.QueryString("look_for")), " ")
```

We populate the strKeyword array with the keyword/s entered on the Form on 'search.htm' page. So strKeyword array contains the list of keywords to search in the database.

```
' Our Connection Object
Set con = CreateObject("ADODB.Connection")
con.Open strDB
```

Create the Connection object and open the database for search.

```
' Our Recordset Object
Set rs = CreateObject("ADODB.Recordset")
rs.CursorLocation = adUseClient
rs.CacheSize = recordsToShow
show = True
```

Create the Recordset object and set the CursorLocation to adUseClient. Why do we do this ? well in order to keep track of the records we have shown to the user and then to show next records to the user on the next page and don't repeat the same records again.

```
' Searching the records for the keywords entered
Select Case UBound(strKeyword)
    Case 0 rs.Open "select * from all_pages where keywords like '%" _
        & strKeyword(0) & "%' order by mydate desc", con
    Case 1 rs.Open "select * from all_pages where keywords like '%" _
        & strKeyword(0) & "%' and keywords like '%" & strKeyword(1) & _
        "%' order by mydate desc", con
    Case 2 rs.Open "select * from all_pages where keywords like '%" _
        & strKeyword(0) & "%' and keywords like '%" & strKeyword(1) & _
        "%' and keywords like '%" & strKeyword(2) & "%' order by _
        mydate desc", con
    Case Else rs.Open "select * from all_pages where keywords like '%" _
        & strKeyword(0) & "%' and keywords like '%" & strKeyword(1) & _
        "%' and keywords like '%" & strKeyword(2) & "%' order by _
        mydate desc", con
End Select
```

We are using classic VBScript select statement to search the Search Engine database for the keywords entered. Depending on the number of keywords entered we use different select statement to search records. Here we are using Recordset object to open the database and use the earlier created Connection as the connection string.

```
' If the returning recordset is not empty
If Not rs.EOF Then

records = records & "found"
rs.PageSize = recordsToShow
totalpages = rs.PageCount
rs.AbsolutePage = currentPage
```

If we have found some matching records in the database then set records to "found". Then we set the PageSize property of Recordset object to the number of records we want to show per page. We also get the number of total pages available to show to the user in the variable totalpages. We then set the AbsolutePage property of Recordset object to the currentPage variable. We do this to keep track of which page we are showing to the user and which page we should be showing.

```
' Showing total number of pages found and the current page number
Response.Write "Displaying Page " & currentPage & " of " & totalPages & "<br>"
Response.Write "Total Records Found : " & rs.RecordCount
Response.Write "<br><br>"
```

We show the user total number of pages of records available and total records found.

```
    ' Showing relevant records
    Do Until rs.EOF
    ' Showing only 10 records at a time
    If n = recordsToShow Then
    Exit Do
    End If

        If InStr(1, urlshown, rs("url"), 1) = 0 Then

        Response.Write "<b><a href=""" & rs("url") & """>" & rs("title") & _
            "</a></b><br>" & vbcrlf
        Response.Write "" & rs("description") & "<br>" & vbcrlf
        Response.Write "URL : " & rs("url") & "</b><br>" & vbcrlf
        Response.Write "Last indexed on : " & rs("mydate") & "<br><br>" & vbcrlf

        n = n + 1
        End If

        urlshown = urlshown & " " & rs("url") & " "
        rs.MoveNext
    Loop
```

Now we start a Do Until loop to show all the available records found to the user while making sure we don't show more records than the number of records specified in recordsToShow variable.

```
    For i = 0 To UBound(strKeyword)
        strKeyword2 = strKeyword(i)
    Next
    look_for = strKeyword2
```

We now store the keywords to search in a variable look_for. We will send the content of this variable to the next page so that when we are moving through the sets of pages showing records, we know which keywords we should be searching.

```
    ' Links to move through the records
    If currentPage > 1 Then
        Response.Write "<a href=""" & Request.ServerVariables("SCRIPT_NAME") & _
            "?currentPage=" & currentPage - 1 & "&look_for=" & look_for & _
            """>Back</a>"
        Else
        Response.Write "<u style=""color : silver;"">Back</u>"
    End If

    Response.Write "        "

    If CInt(currentPage) <> CInt(totalPages) Then
        Response.Write "<a href=""" & Request.ServerVariables("SCRIPT_NAME") & _
            "?currentPage=" & currentPage + 1 & "&look_for=" & look_for & _
            """>Next</a>"
        Else
        Response.Write "<u style=""color : silver;"">Next</u>"
    End If
```

We now create 'Back' and 'Next' links to show accordingly if records are available to move back and forth.

```
    End If
```

```
     ' Done. Now release Objects
     con.Close
     Set con = Nothing
     Set rs = Nothing

   If InStr(1, records, "found", 1) = 0 Then
   Response.Write "<b>Sorry, no matching record was found.</b>" & vbcrlf
   End If
%>
   </body></html>
```
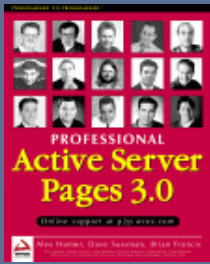
In the end release the objects created during the execution of the page. If no matching records were found then we inform the user that no matching records were found.

On the next page we will create the 'editme.asp' page.

[Next ->](#)

© 1999 - 2001 [Stardeveloper.com](#).

STARDEVELOPER
.COM

7/12/2001

Quick Links :   [home](#) | [resources](#) | [book store](#) | [forum](#) | [whats new](#) | [advertise](#) | [contact us](#)

Active Server Pages 3.0

[Sample Chapter](#)

News Letter

www.wantjava.com

FREE WEB RESOURCES
CLICK HERE

'editme.asp' Include File

Creating 'editme.asp' Include File :

Create a new ASP page and name it as 'editme.asp'. Copy paste the following code into it :

```
<%
  ' Path to Database
  strDB = "Provider=Microsoft.Jet.OLEDB.4.0; Data Source=" & _
      Server.MapPath("directory.mdb")
%>
```

Very simple to understand. strDB contains the database string required to connect to the database. By default 'directory.mdb' will be present in the same directory where you are keeping other Search Engine files. If you want to put 'directory.mdb' database else where then you should change the location to database in 'editme.asp' page as well e.g. if you change database path to http://www.yoursite.com/personal/db/directory.mdb then you should put the following code in place of the above :

```
<%
  ' Path to Database
  strDB = "Provider=Microsoft.Jet.OLEDB.4.0; Data Source=" & _
      Server.MapPath("/personal/db/directory.mdb")
%>
```

We summarize what we did on the next page and then end this tutorial there.

Please use a Java Script enabled browser.

[Next ->](#)

STARDEVELOPER
.COM

7/12/2001

Quick Links : home | resources | book store | forum | whats new | advertise | contact us

Navigation

Overview
Access Database
Indexing using
HTTP protocol
Indexing using
FilySystemObject
Deleting URLs from
Database
HTML Search
Engine Form
Searching Records in
the Database
Include File
Conclusion

Sample Chapter

News Letter

www.
wantjava
.com
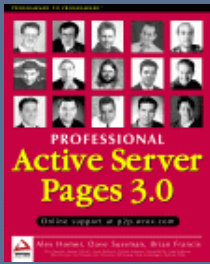
FREE WEB
RESOURCES
CLICK HERE

Conclusion

Conclusion :

In this long article / tutorial we created a very useful search engine from scratch. We learned how simple VBScript methods can be used to create effective web applications. We also saw that dealing with databases is very easy in ASP.

Note before installing the search engine, read the 'readme.htm' file accompanying the download. Enjoy !

- Download complete working Search Engine.
- Tell a friend about this article / tutorial.

Please use a Java Script enabled browser.

© 1999 - 2001 <u>Stardeveloper.com</u>.