

# Simple ASP Authentication System

by Manohar Kamath

Feb 15th, 1998

This article describes how you can secure ASP applications using simple, but quite effective authentication schemes. This article uses a very simple way to achieve this. Just follow the steps and you have a secure login system.

## Step 1: Create a table of users



tUsers	
UserName	
UserPassword	

Just create a simple table of user logins and passwords. I have included a database userinfo.mdb with this example, which contains a sample table tUsers. tUsers has two fields - Username and UserPassword. Username is the primary key.

Download and copy this database on your hard disk.

## Step 2: Set the default authentication status

This you do in the global.asa file. All you have to do is, set a session variable to a default "not authenticated" status.

Why? Because, when a users first come into the application, they are not valid until you have checked their "credentials." The default status makes sure that everyone has to go through the front door.

In global.asa file, within the Session\_OnStart event, write this code

```
<SCRIPT LANGUAGE=VBScript RUNAT=Server>
SUB Session_OnStart
' This is the default authentication status
Session("Authenticated") = 0
END SUB
</SCRIPT>
```

The authentication status is the most important thing to keep in mind, so don't forget this.

## Step 3: Create a login page

This is an ASP page, with just HTML in it. Call it say login.asp. For your convenience, here is the sample code:

```
<HTML>
<BODY BGCOLOR=FFFFFF>
```

```
<FORM ACTION="verify.asp" METHOD=POST>
Name:
<INPUT TYPE=TEXT SIZE=20 NAME=USERNAME>

Password:
<INPUT TYPE=PASSWORD SIZE=20 NAME=USERPASSWORD>
<INPUT TYPE=SUBMIT VALUE="Login Now">
</BODY>
</HTML>
```

It contains a form with 2 INPUT elements. These elements are used to collect the user name and password of the user. This information we POST to verify.asp where we verify if the user is valid or not.

#### **Step 4: Create the system DSN for the database**

In order to access the userinfo.mdb, we need to create a system DSN in ODBC. If you are familiar with ASP, you can choose your own DSN scheme. To create a system DSN, do the following:

- Open the Control panel of your machine (in Start ..Settings menu in Windows 95/NT)
- Click on "ODBC"
- Click on "System DSN" tab
- Click "Add". Choose the "Microsoft Access Database Driver", and click "Finish"
- Give the DSN a name, say "LoginDSN" In "Database" settings, click "Select" and point to the userinfo.mdb on your hard disk.
- Click OK

This sets up a system DSN named "LoginDSN" on your machine. This will point to the userinfo.mdb on the hard disk.

#### **Step 5: Create an authentication page**

This is the verify.asp page we saw in step 3. In this page, we check for valid users. We get the user information from the login.asp (remember the form elements?)

Our intent is

- Check for valid users and set the authentication status accordingly
- If the user is valid, the authentication status is 1
- If the user is invalid, the authentication status is 0

The code for verify.asp is as shown below. You can modify it accordingly.

```

<%
' Create a command object. This object serves to run our queries
Set Cm = Server.CreateObject("ADODB.Command")

' Specify the system DSN path
Cm.ActiveConnection = "LoginDSN"

' Now it's time for the query. We need to check the user information
' against the table tUsers
Cm.CommandText = "SELECT * FROM tUsers WHERE " & _
  "UserName=" & Request.Form("UserName") & " AND " & _
  "UserPassword=" & Request.Form("UserPassword") & " "

' Set the query type. 1 means it is a SQL statement
Cm.CommandType = 1

' Retrieve the results in a recordset object
Set Rs = Cm.Execute

' We now check if the user is valid. If user is valid, the recordset MUST
' haverecord. Otherwise it is empty. If user exists, we set authentication
' status to 1 and send the user to appropriate page, say welcome.asp.
' Else send the user back to login.asp
If Rs.EOF Then
  Session("Authenticated") = 0
  Response.Redirect ("login.asp")
Else
  Session("Authenticated") = 1
  Response.Redirect ("welcome.asp")
End If
%>

```

## Step 6: Check the authentication status

This is the important piece of our system. We must check the authentication status on EACH ASP PAGE that we want to be secured. This is simple to do. Just check if the authentication status is 1, if not send the user back to login.asp. The sample code is

```

<%
If Session("Authenticated") = 0 Then
  Response.Redirect ("Login.asp")
End If
%>

```

Alternatively, you can copy this code into a file, say check.inc, and include the following code on top of your files instead.

```
<!-- #include file="check.inc" -->
```

As I mentioned before, this code needs to go on TOP of each page that you want to protect.

The above 6 steps help you to create a simple authentication system. Remember that this system protects ONE virtual directory and not the whole web site. You need to create one for each virtual path you want to secure.

Also, the above system is targeted towards new users. The database and the code is kept simple so you can learn from it. The entire system can be downloaded from this site. The zipped file contains the database and all the files.