

ASP Database Access

EIW - ASP Database Access

1

Server Object: ActiveX Data Object (ADO)

- PWS or ISS provides database connectivity via an ActiveX Data Object.
- You can still refer to the ODBC DSN when connecting to the Database
 - we did this with Perl
- You can write code that doesn't care whether the database is Access, SQL Server, etc.

EIW - ASP Database Access

2

Creating a database with Access and an ODBC DSN

- Refer to the Perl Database lecture notes for the details...
- Database access is similar to what we did in Perl, but we use a VBScript ADO object instead of Win32::ODBC.
- Database commands are SQL

EIW - ASP Database Access

3

Connecting to a Database

- Create an **ADODB.Connection** object:

```
Dim dbconn
Set dbconn=Server.CreateObject("ADODB.Connection")
```

Need the "set" keyword here since we are assigning an object to a VBScript variable

EIW - ASP Database Access

4

DB Connection Object

- This object represents an actual connection to a database.
 - It can be expensive to open/close database connections.
 - It is possible to tell the web server to do *connection pooling*:
 - database holds a number of open connections and allows various ASP documents to use them.
 - You need to change the ODBC DSN properties to enable connection pooling
 - You don't need to change your ASP pages to take advantage of this.

EIW - ASP Database Access

5

Connecting to a DSN

- Use the Open method:

```
Dim dbconn
Set dbconn=Server.CreateObject("ADODB.Connection")
dbconn.open("eiw")
```

There are other optional arguments (can specify id,password)

You specify the DSN here

EIW - ASP Database Access

6

SQL and RecordSet Objects

- The **ADODB.Connection** object has a method named **Execute**
 - give it an **SQL** command (as a parameter)
 - It creates a **RecordSet** object
- The **RecordSet** object contains the result of the **SQL** query.

EIW - ASP Database Access

7

Execute Example

```
Dim dbconn
Set dbconn=Server.CreateObject("ADODB.Connection")
dbconn.open("eiw")

Set result=dbconn.Execute("SELECT * FROM Password")
```

Resulting RecordSet object This is the SQL Query

EIW - ASP Database Access

8

RecordSet Objects

- A **RecordSet** object holds a collection of records (from the database) and has some methods:
 - **EOF ()** : boolean that indicates there are no more records.
 - **MoveNext ()** : moves to the next record
 - **MovePrevious ()** : move to prev. record
 - **MoveFirst ()** : move to the first record
 - Lots more... (check an ASP reference for the entire list)

EIW - ASP Database Access

9

Getting at DB Field Values

- Use the RecordSet as a collection:

```
Set result=dbconn.Execute("SELECT * FROM Password")
' grab the first name,password
name=result("name")
pass=result("password")
' move to the next record
result.MoveNext
```

← name and password
are fields (columns)
in the Password table

EIW - ASP Database Access

10

Looping over all records returned

```
Set result=dbconn.Execute("SELECT * FROM Password")
Do Until result.EOF()
  name=result("name")
  pw=result("password")
  ' do something with name,password..
  result.MoveNext()
Loop
```

←

Note: Leave this line out and you will hang PWS
(and you will need to reboot).
Hopefully IIS behaves better !?!

EIW - ASP Database Access

11

Database Updates

- Change existing records
 - Use RecordSet Update method
- Add new records
 - Use Connection.Execute method
 - or
 - Use RecordSet Addnew and Update methods

EIW - ASP Database Access

12

Login System in ASP

- Same Database as we built in the Perl-Database lecture.
 - Table named "Password"
 - Fields: "name" and "password"
- Need to know the ODBC DSN
 - DSN is "eiw" in the code shown here

Login Form

```
<FORM ACTION=index.asp>
Name: <INPUT TYPE=TEXT NAME=name><BR>
Password:<INPUT TYPE=password NAME=pw><BR>
<INPUT TYPE=submit Name=operation value="Login">
</FORM>
```

Create Account Form

```
<FORM ACTION=index.asp>
Full Name: <INPUT TYPE=TEXT Name=fullname><BR>
ID:<INPUT TYPE=TEXT NAME=name><BR>
Password: <INPUT TYPE=password NAME=pw><BR>
Credit Card Number:<INPUT TYPE=text Name=cc><BR>
<INPUT TYPE=submit Name=operation value="Create">
</FORM>
```

Code Overview

- If operation is Login
 - validate name,pw.
- If operation is Create
 - make sure we got a name,pw
 - make sure name is not already used
 - create new record in database

EIW - ASP Database Access

16

ValidLogin function

```
Function ValidLogin (name ,pw)
Dim DBConn
Set DBConn=Server.CreateObject("ADODB.Connection")
DBConn.Open("eiw")
Set res = DBConn.Execute("SELECT * FROM Password")
ValidLogin=False
Do Until res.EOF
  If res("Name")=name And res("Password")=pw Then
    ValidLogin=True
    Exit Do
  End If
  res.MoveNext
Loop
End Function
```

EIW - ASP Database Access

17

ValidLogin function

```
Function ValidLogin (name ,pw)
Dim DBConn
Set DBConn=Server.CreateObject("ADODB.Connection")
DBConn.Open("eiw")
Set res = DBConn.Execute("SELECT * FROM Password")
ValidLogin=False
Do Until res.EOF
  If res("name")=name And res("password")=pw Then
    ValidLogin=True
    Exit Do
  End If
  res.MoveNext
Loop
End Function
```

EIW - ASP Database Access

18

A Better ValidLogin

```
Function ValidLogin(name,pw)
  Dim DBConn
  Set DBConn=Server.CreateObject("ADODB.Connection")
  DBConn.Open("eiw")
  Set res = DBConn.Execute _ ← VBScript line continuation
    ("SELECT * FROM Password WHERE name = '&name&'")
  If res.EOF Then
    ValidLogin=False
  ElseIf res("password") = pw Then
    ValidLogin=True
  Else
    ValidLogin=False
  End If
End Function
```

operation="Login"

```
If Request("operation")="Login" And _
Request("name")<>" And Request("pw")<>" Then
  ' check for valid name,pw
  If ValidLogin(Request("name"),Request("pw")) Then%>
    <H2>You are logged in <%=Request("name")%></H2>
  <%Else%>
    <H2>Invalid Name or Password</H2>
  End If
Else %>
  <H2>You have to enter a name and password</H2>
<%End If
```

Creating a new record

- SQL "INSERT" command:

```
INSERT into table (field1, field2, ...)
VALUES (val1, val2, ...)
```

- For string type fields the values must be in single quotes!

New password record

Table Name Field Names

```

INSERT INTO Password (name, password)
VALUES ('george', 'monkey')

```

Field Values

EIW - ASP Database Access 22

NewAccount subroutine

```

Sub NewAccount(name,pw)
  Set DBConn=Server.CreateObject("ADODB.Connection")
  DBConn.Open("eiw")
  DBConn.Execute("INSERT INTO password (Name, Password)
VALUES ('&name&', '&pw &');")
  ' need to do some error checking !
End Sub

```

This is all one line

EIW - ASP Database Access 23

There are other ways...

- Instead of the SQL INSERT command, you can use ADO (RecordSet) object methods to create new records (or change existing records):

```

Sub NewAccount(name,pw)
  Set DBConn=Server.CreateObject("ADODB.Connection")
  DBConn.Open("eiw")
  Set RS = Server.CreateObject("ADODB.RecordSet")
  RS.Open "password",DBConn,adOpenDynamic,adLockOptimistic
  RS.AddNew                      ← Creates new record
  RS("Name")=name                ← sets values for new record
  RS("Password")=pw              ←
  RS.Update                      ← updates the database
End Sub

```

EIW - ASP Database Access 24

Make sure name is not already used.
Can use function **GetPassword**

```
Function GetPassword(name)
Dim DBConn
Set DBConn=Server.CreateObject("ADODB.Connection")
DBConn.Open("eivw")
Set res = DBConn.Execute("SELECT * FROM Password
Where Name = '" & name & "'")
If res.EOF Then
    GetPassword=""
Else
    GetPassword=res("password")
End If
End Function
```

EIW - ASP Database Access

25

operation="Create"

```
If Request("operation")="Create" Then
If Request("name") = "" OR Request("pw") = "" Then %>
<H2>Invalid name or password</H2>
<%ElseIf GetPassword(Request("name"))<>"" Then %>
<H2>That name is already taken</H2>
<%Else
Call NewAccount(Request("name"),Request("pw")) %>
<H2>Your account has been created, log in</H2>
<%End If
End If
```

EIW - ASP Database Access

26

Record Updates

- We may want to allow people to change their password.
 - we need to update a single record.
- General Strategies
 - Use SQL
 - or
 - Find the record (with a RecordSet Object)
 - Use the RecordSet Update Method

EIW - ASP Database Access

27

Change Password Form

```
<FORM ACTION=index.asp>
ID:<INPUT TYPE=TEXT NAME=name><BR>
Old Password: <INPUT TYPE=password NAME=opw><BR>
New Password: <INPUT TYPE=password NAME=opw><BR>
<INPUT TYPE=submit Name=operation value="Change">
</FORM>
```

EIW - ASP Database Access

28

SQL Updates

```
UPDATE table SET fname1=newval1
fname2=newval2 ... WHERE ...
```

```
UPDATE Password SET
password='blah' WHERE
name='Dave'
```

EIW - ASP Database Access

29

ChangePassword subroutine

```
Sub ChangePassword(name, pw)
Set DBConn=Server.CreateObject("ADODB.Connection")
DBConn.Open("eiw")
DBConn.Execute("UPDATE password Set Password = '" &
pw & "' WHERE Name='" & name & "'")
' need to do some error checking !
End Sub
```

This is all one line

EIW - ASP Database Access

30

ChangePassword using RecordSet

```
Sub ChangePassword(name, pw)
Set DBConn=Server.CreateObject("ADODB.Connection")
DBConn.Open("eiw")
Set RS = Server.CreateObject("ADODB.RecordSet")
RS.Open "SELECT * FROM password WHERE Name =
'&name&'",DBConn,adOpenDynamic,adLockOptimistic
RS("password")=pw
RS.Update
End Sub
```

This is all one line

EIW - ASP Database Access

31

operation="Change"

```
If Request("operation") = "Change" Then
'check for valid name
If Request("name") = "" OR Request("pw") = "" Then %>
<H2>Invalid name or password</H2>
<%ElseIf GetPassword(Request("name"))="" Then %>
<H2>No such name (ID) found</H2>
<%ElseIf Not ValidLogin(Request("name"),Request("opw"))
Then %>
<H2>Bad (old) password</H2>
<%Else
Call ChangePassword(Request("name"),Request("pw")) %>
<H2>Your Password has been changed.</H2>
<%End If
End If
```

EIW - ASP Database Access

32

Important Issues

- ADO constants like `adOpenDynamic` and `adLockOptimistic` are defined in a file named `adovbs.inc`
 - located at
`C:\Program Files\Common Files\SYSTEM\ADO`
- Demo code uses SSI include directive:
`<!--#include file="adovbs.inc" -->`

EIW - ASP Database Access

33

ADO RecordSet Object vs. SQL

- Learn only ADO object methods
 - simpler syntax in many cases
 - You can be a MS/VBScript Guru
- Learn SQL:
 - works with many databases, languages
 - If you do anything fancy you probably need to use SQL.

Possible E-Commerce Database

Table: User
user_id
name
password

Table: Cart
user_id
product_id
qty

Table: Product
product_id
name
description
image (URL)
price
qty

Table: Purchase
user_id
product_id
qty
date

Some Possible Queries

- Get user information given a name
- Get all items in a shopping cart given a user_id.
- Get all items in user purchase history
- Get all products with "memory" in the name.

Get user information given a name

```
SELECT * FROM User WHERE name='fred'
```

Get all fields Table name Restricts the records found

Get all items in a shopping cart given a user_id.

```
SELECT product_id, qty FROM Cart WHERE user_id=107
```

Get just these fields Table name Restricts the records found

Get all items in user purchase history

- We need to match entries in Purchase that have a specific user_id.
- We might want information about each item as well.

Get all items in user purchase history

```
SELECT * FROM Purchase,Product
WHERE user_id=107 AND Purchase.product_id
= Product.product_id
```

EIW - ASP Database Access

40

Get all products with "memory" in the name.

- We need to use the SQL "LIKE" keyword.
 - % means "any string of zero or more characters"
 - In some SQL version you use "*"
 - ? means "any single character"
 - [] is like regexp character class
 - [!] is like negated character class

EIW - ASP Database Access

41

ASP Applications

- An *Application* is defined as all the documents within a virtual folder (and any folders it contains).
- You can force the issue by creating a **Global.asa** file and put it in a folder.
 - everything in the folder is *an application*.
 - Applications share application variables...

EIW - ASP Database Access

42

Global.asa

- asa is "Active Server Application"
- You can define 4 subroutines:
 - `Application_OnStart`
 - `Application_OnEnd`
 - `Session_OnStart`
 - `Session_OnEnd`

EIW - ASP Database Access

43

Example

```
Sub Application_OnStart
    Application("ActiveUsers")=0
End Sub

Sub Session_OnStart
    Application("ActiveUsers")=Application("ActiveUsers")+1
End Sub

Sub Session_OnEnd
    Application("ActiveUsers")=Application("ActiveUsers")-1
End Sub
```

EIW - ASP Database Access

44
