For a long time, ODBC (Open Database Connectivity) has been used as a standard programming language interface that allows an application access to a variety of data sources. Using ODBC, access is provided to any data source, either local or remote, as long as an appropriate ODBC driver is available, ODBC drivers are available for a wide variety of data source such as Microsoft SQL Server and Microsoft Access.

While ODBC proved to be an efficient way to access relational data source while maintaining interoperability in database applications, there remained a few problems with using the ODBC API to access data sources. The ODBC API proved to be too complex for most developers to develop truly interoperable applications.

Microsoft introduced OLE DB as an enhancement to ODBC and eliminate any and all biases to specific data-storage formats. OLE DB represents a COM interface directly to the target database. OLE DB provides a component solution for database access to heterogeneous data sources. Since OLE DB is a C++ API, it is not directly accessible to many other development environments such as Visual Basic and Internet platforms. For these environments, Microsoft has provided a set of programmable objects known as the Active X data objects (ADO) that allow access to OLE DB.

ADO was introduced by Microsoft in March, 1997 as part of Visual Studio 971. It was originally intended as a method through which ASPs running on is could access OLE DB data sources. ADO is also usable in Visual Basic, Visual C++ and Visual J++ Java applications.

The most popular implementation of ADO today can be found in ASP. ASP enables connecting to external data sources through ADO. OLE DB and ADO are not intended to replace ODBC. But if your application accesses data from data sources ranging from mainframes to desktops or if you are building a Web application designed to provide a user with an interactive experience with a data source, OLE DB and ADO are just the right solution for you.

What is ADO?

* ADO is a Microsoft technology
* ADO stands for **A**ctiveX **D**ata **O**bjects
* ADO is a Microsoft Active-X component
* ADO is automatically installed with Microsoft IIS
* ADO is a programming interface to access data in a database

**Advantages of ADO:**

The primary advantages of ADO are:

• Ease of use

• Speed

• Minimal memory overhead

• Unbiased access to relational and non-relational data sources

**Features of ADO:**

ADO includes the following key features:

• Support for batch updating. A number of record updates are cached and transmitted at one time.

• Support for all types of cursors including forward-only, key set, dynamic and static type cursors.

• Support for server-side stored procedures.

• Support for queries that return multiple record sets.

• Support for query goals such as limits on the number of records returned, active filtering of the records returned and prepared statements.

Application

Web server Application

ADO

OLE DB

ODBC

OLEDB data source

ODBC data source

Fig: ADO application configuration.

**Accessing a Database from an ASP Page:**

The common way to access a database from inside an ASP page is to:

1. Create an ADO connection to a database
2. Open the database connection
3. Create an ADO record set
4. Open the record set
5. Extract the data you need from the record set
6. Close the record set
7. Close the connection

**What is DSN?**

**Data Source Name**

Data Source Name provides connectivity to a database through an ODBC driver. The DSN contains database name, directory, database driver, UserID, password, and other information. Once you create a DSN for a particular database, you can use the DSN in an application to call information from the database.

**There are three types of DSNs:**

(1) System DSN— can be used by anyone who has access to the machine. DSN info is stored in the registry.

(2) User DSN— created for a specific user. Also stored in the registry.

(3) File DSN— DSN info is stored in a text file with .DSN extensions.

## Create a DSN-less Database Connection:

The easiest way to connect to a database is to use a DSN-less connection. A DSN-less connection can be used against any Microsoft Access database on your web site. If you have a database called "northwind.mdb" located in a web directory like "c:/webdata/", you can connect to the database with the following ASP code:

<%  
set conn=Server.CreateObject("ADODB.Connection")  
conn.Provider="Microsoft.Jet.OLEDB.4.0"

conn.Open "c:/webdata/northwind.mdb"

%>

**An ODBC Connection to an MS Access Database:**

Here is how to create a connection to a MS Access Database:

1. Open the **ODBC** icon in your Control Panel.
2. Choose the **System DSN** tab.
3. Click on **Add**in the System DSN tab.
4. **Select** the Microsoft Access Driver. Click **Finish.**
5. In the next screen, click **Select** to locate the database.
6. Give the database a **D**ata **S**ource **N**ame (DSN).
7. Click **OK**.

**For Access database**:-

**With native OLE DB Provider** (preferred):

Provider=Microsoft.Jet.OLEDB.4.0; Data Source=c:\domains\yourdomain.com\db\yourdatabase.mdb

**Using ODBC connection without specifying a DSN**:

Driver={Microsoft Access Driver (\*.mdb)}; DBQ=c:\domains\yourdomain.com\db\yourdatabase.mdb

**Microsoft SQL Server:**

If you are connecting to Microsoft SQL Server then replace Admin and test with valid SQL Server Login Name and Password.

Connecting to database with DSN, after a System DSN is created you can open connection to that database in an ASP page

with this simple code.

<%

Set MyConn = Server.CreateObject(“ADODB.Connection”)

MyConn.Open “DSN=MyDSN;UID=Admin;PWD=Test”

%>

**ADO Connection Object:**

The ADO Connection Object is used to create an open connection to a data source. Through this connection, you can access and manipulate a database. If you want to access a database multiple times, you should establish a connection using the Connection object. You can also make a connection to a database by passing a connection string via a Command or Record set object. However, this type of connection is only good for one specific, single query.

**Syntax:**

set objConnection=Server.CreateObject("ADODB.connection")

## ADO Command Object:

The ADO Command object is used to execute a single query against a database. The query can perform actions like creating, adding, and retrieving, deleting or updating records. If the query is used to retrieve data, the data will be returned as a Record Set object. This means that the retrieved data can be manipulated by properties, collections, methods, and events of the Record set object. The major feature of the Command object is the ability to use stored queries and procedures with parameters.

### **Syntax:**

set objCommand=Server.CreateObject("ADODB.command")

# **ADO Record set:**

To be able to read database data, the data must first be loaded into a record set.

## Create an ADO Table Record set:

After an ADO Database Connection has been created, as demonstrated in the previous chapter, it is possible to create an ADO Record set.

<%  
set conn=Server.CreateObject("ADODB.Connection")  
conn.Provider="Microsoft.Jet.OLEDB.4.0"  
conn.Open "c:/webdata/northwind.mdb"  
  
set rs=Server.CreateObject("ADODB.recordset")  
rs.Open "Customers", conn  
%>

## Add a Record to a Table in a Database

## Guestbook\_form.html

<html>

<head>

<title>Guestbook Form</title>

</head>

<body bgcolor="white" text="black">

<!-- Begin form code -->

<form name="form" method="post" action="add\_to\_guestbook.asp">

Name: <input type="text" name="name" maxlength="20">

<br>

Comments: <input type="text" name="comments" maxlength="60">

<input type="submit" name="Submit" value="Submit">

</form>

<!-- End form code -->

</body></html>

**guestbook.asp**

<%

'Dimension variables

Dim adoCon 'Holds the Database Connection Object

Dim rsAddComments 'Holds the recordset for the new record to be added to the database

Dim strSQL 'Holds the SQL query for the database

'Create an ADO connection odject

Set adoCon = Server.CreateObject("ADODB.Connection")

'Set an active connection to the Connection object using a DSN-less connection

adoCon.Open "DRIVER={Microsoft Access Driver (\*.mdb)}; DBQ=" & Server.MapPath("guestbook.mdb")

'Set an active connection to the Connection object using DSN connection

'adoCon.Open "DSN=guestbook"

'Create an ADO recordset object

Set rsAddComments = Server.CreateObject("ADODB.Recordset")

'Initialise the strSQL variable with an SQL statement to query the database

strSQL = "SELECT tblComments.Name, tblComments.Comments FROM tblComments;"

'Set the cursor type we are using so we can navigate through the recordset

rsAddComments.CursorType = 2

'Set the lock type so that the record is locked by ADO when it is updated

rsAddComments.LockType = 3

'Open the tblComments table using the SQL query held in the strSQL varaiable

rsAddComments.Open strSQL, adoCon

'Tell the recordset we are adding a new record to it

rsAddComments.AddNew

'Add a new record to the recordset

rsAddComments.Fields("Name") = Request.Form("name")

rsAddComments.Fields("Comments") = Request.Form("comments")

'Write the updated recordset to the database

rsAddComments.Update

'Reset server objects

RsAddComments. Close

Set rsAddComments = Nothing

Set adoCon = Nothing

'Redirect to the guestbook.asp page

Response. Redirect "guestbook.asp"

**Guestbook.asp**

<html>

<head>

<title>Guestbook</title>

</head>

<body bgcolor="white" text="black">

<%

'Dimension variables

Dim adoCon 'Holds the Database Connection Object

Dim rsGuestbook 'Holds the recordset for the records in the database

Dim strSQL 'Holds the SQL query for the database

'Create an ADO connection object

Set adoCon = Server.CreateObject("ADODB.Connection")

'Set an active connection to the Connection object using a DSN-less connection

adoCon.Open "DRIVER={Microsoft Access Driver (\*.mdb)}; DBQ=" & Server.MapPath("guestbook.mdb")

'Set an active connection to the Connection object using DSN connection

'adoCon.Open "DSN=guestbook"

'Create an ADO recordset object

Set rsGuestbook = Server.CreateObject("ADODB.Recordset")

'Initialise the strSQL variable with an SQL statement to query the database

strSQL = "SELECT tblComments.Name, tblComments.Comments FROM tblComments;"

'Open the recordset with the SQL query

rsGuestbook.Open strSQL, adoCon

'Loop through the recordset

Do While not rsGuestbook.EOF

'Write the HTML to display the current record in the recordset

Response.Write ("<br>")

Response.Write (rsGuestbook("Name"))

Response.Write ("<br>")

Response.Write (rsGuestbook("Comments"))

Response.Write ("<br>")

'Move to the next record in the recordset

rsGuestbook.MoveNext

Loop

'Reset server objects

rsGuestbook.Close

Set rsGuestbook = Nothing

Set adoCon = Nothing

%>

</body>

</html>%>

**Updating a record using ADO**

**Update\_select.asp**

<html>

<head>

<title>Update Entry Select</title>

</head>

<body bgcolor="white" text="black">

<%

'Dimension variables

Dim adoCon 'Holds the Database Connection Object

Dim rsGuestbook 'Holds the recordset for the records in the database

Dim strSQL 'Holds the SQL query for the database

'Create an ADO connection odject

Set adoCon = Server.CreateObject("ADODB.Connection")

'Set an active connection to the Connection object using a DSN-less connection

adoCon.Open "DRIVER={Microsoft Access Driver (\*.mdb)}; DBQ=" & Server.MapPath("guestbook.mdb")

'Set an active connection to the Connection object using DSN connection

'adoCon.Open "DSN=guestbook"

'Create an ADO recordset object

Set rsGuestbook = Server.CreateObject("ADODB.Recordset")

'Initialise the strSQL variable with an SQL statement to query the database

strSQL = "SELECT tblComments.\* FROM tblComments;"

'Open the recordset with the SQL query

rsGuestbook.Open strSQL, adoCon

'Loop through the recordset

Do While not rsGuestbook.EOF

'Write the HTML to display the current record in the recordset

Response.Write ("<br>")

Response.Write ("<a href=""update\_form.asp?ID=" & rsGuestbook("ID\_no") & """>")

Response.Write (rsGuestbook("Name"))

Response.Write ("</a>")

Response.Write ("<br>")

Response.Write (rsGuestbook("Comments"))

Response.Write ("<br>")

'Move to the next record in the recordset

rsGuestbook.MoveNext

Loop

'Reset server objects

rsGuestbook.Close

Set rsGuestbook = Nothing

Set adoCon = Nothing

%>

</body>

</html>

**Update\_form.asp**

<%

'Dimension variables

Dim adoCon 'Holds the Database Connection Object

Dim rsGuestbook 'Holds the recordset for the record to be updated

Dim strSQL 'Holds the SQL query for the database

Dim lngRecordNo 'Holds the record number to be updated

'Read in the record number to be updated

lngRecordNo = CLng(Request.QueryString("ID"))

'Create an ADO connection odject

Set adoCon = Server.CreateObject("ADODB.Connection")

'Set an active connection to the Connection object using a DSN-less connection

adoCon.Open "DRIVER={Microsoft Access Driver (\*.mdb)}; DBQ=" & Server.MapPath("guestbook.mdb")

'Set an active connection to the Connection object using DSN connection

'adoCon.Open "DSN=guestbook"

'Create an ADO recordset object

Set rsGuestbook = Server.CreateObject("ADODB.Recordset")

'Initialise the strSQL variable with an SQL statement to query the database

strSQL = "SELECT tblComments.\* FROM tblComments WHERE ID\_no=" & lngRecordNo

'Open the recordset with the SQL query

rsGuestbook.Open strSQL, adoCon

%>

<html>

<head>

<title>Guestbook Update Form</title>

</head>

<body bgcolor="white" text="black">

<!-- Begin form code -->

<form name="form" method="post" action="update\_entry.asp">

Name: <input type="text" name="name" maxlength="20" value="<% = rsGuestbook("Name") %>">

<br>

Comments: <input type="text" name="comments" maxlength="60" value="<% = rsGuestbook("Comments") %>">

<input type="hidden" name="ID\_no" value="<% = rsGuestbook("ID\_no") %>">

<input type="submit" name="Submit" value="Submit">

</form>

<!-- End form code -->

</body>

</html>

<%

'Reset server objects

rsGuestbook.Close

Set rsGuestbook = Nothing

Set adoCon = Nothing

%>

**Update\_entry.asp**

<%

'Dimension variables

Dim adoCon 'Holds the Database Connection Object

Dim rsUpdateEntry 'Holds the recordset for the record to be updated

Dim strSQL 'Holds the SQL query for the database

Dim lngRecordNo 'Holds the record number to be updated

'Read in the record number to be updated

lngRecordNo = CLng(Request.Form("ID\_no"))

'Create an ADO connection odject

Set adoCon = Server.CreateObject("ADODB.Connection")

'Set an active connection to the Connection object using a DSN-less connection

adoCon.Open "DRIVER={Microsoft Access Driver (\*.mdb)}; DBQ=" & Server.MapPath("guestbook.mdb")

'Set an active connection to the Connection object using DSN connection

'adoCon.Open "DSN=guestbook"

'Create an ADO recordset object

Set rsUpdateEntry = Server.CreateObject("ADODB.Recordset")

'Initialise the strSQL variable with an SQL statement to query the database

strSQL = "SELECT tblComments.\* FROM tblComments WHERE ID\_no=" & lngRecordNo

'Set the cursor type we are using so we can navigate through the recordset

rsUpdateEntry.CursorType = 2

'Set the lock type so that the record is locked by ADO when it is updated

rsUpdateEntry.LockType = 3

'Open the tblComments table using the SQL query held in the strSQL varaiable

rsUpdateEntry.Open strSQL, adoCon

'Update the record in the recordset

rsUpdateEntry.Fields("Name") = Request.Form("name")

rsUpdateEntry.Fields("Comments") = Request.Form("comments")

'Write the updated recordset to the database

rsUpdateEntry.Update

'Reset server objects

rsUpdateEntry.Close

Set rsUpdateEntry = Nothing

Set adoCon = Nothing

'Return to the update select page incase another record needs deleting

Response.Redirect "update\_select.asp"

%>

**Deleting a record using ADO:**

**Delete\_select.asp**

<html>

<head>

<title>Delete Entry Select</title>

</head>

<body bgcolor="white" text="black">

<%

'Dimension variables

Dim adoCon 'Holds the Database Connection Object

Dim rsGuestbook 'Holds the recordset for the records in the

database

Dim strSQL 'Holds the SQL query for the database

'Create an ADO connection odject

Set adoCon = Server.CreateObject("ADODB.Connection")

'Set an active connection to the Connection object using a DSN-less connection

adoCon.Open "DRIVER={Microsoft Access Driver (\*.mdb)}; DBQ=" & Server.MapPath

("guestbook.mdb")

'Set an active connection to the Connection object using DSN connection

'adoCon.Open "DSN=guestbook"

'Create an ADO recordset object

Set rsGuestbook = Server.CreateObject("ADODB.Recordset")

'Initialise the strSQL variable with an SQL statement to query the database

strSQL = "SELECT tblComments.\* FROM tblComments;"

'Open the recordset with the SQL query

rsGuestbook.Open strSQL, adoCon

'Loop through the recordset

Do While not rsGuestbook.EOF

'Write the HTML to display the current record in the recordset

Response.Write ("<br>")

Response.Write ("<a href=""delete\_entry.asp?ID=" & rsGuestbook("ID\_no") &

""">")

Response.Write (rsGuestbook("Name"))

Response.Write ("</a>")

Response.Write ("<br>")

Response.Write (rsGuestbook("Comments"))

Response.Write ("<br>")

'Move to the next record in the recordset

rsGuestbook.MoveNext

Loop

'Reset server objects

rsGuestbook.Close

Set rsGuestbook = Nothing

Set adoCon = Nothing

%>

</body>

</html>

**Delete\_entry.asp**

<%

'Dimension variables

Dim adoCon 'Holds the Database Connection Object

Dim rsDeleteEntry 'Holds the recordset for the record to be deleted

Dim strSQL 'Holds the SQL query for the database

Dim lngRecordNo 'Holds the record number to be deleted

'Read in the record number to be deleted

lngRecordNo = CLng(Request.QueryString("ID"))

'Create an ADO connection odject

Set adoCon = Server.CreateObject("ADODB.Connection")

'Set an active connection to the Connection object using a DSN-less connection

adoCon.Open "DRIVER={Microsoft Access Driver (\*.mdb)}; DBQ=" & Server.MapPath

("guestbook.mdb")

'Set an active connection to the Connection object using DSN connection

'adoCon.Open "DSN=guestbook"

'Create an ADO recordset object

Set rsDeleteEntry = Server.CreateObject("ADODB.Recordset")

'Initialise the strSQL variable with an SQL statement to query the database

strSQL = "SELECT tblComments.\* FROM tblComments WHERE ID\_no=" & lngRecordNo

'Set the lock type so that the record is locked by ADO when it is deleted

rsDeleteEntry.LockType = 3

'Open the recordset with the SQL query

rsDeleteEntry.Open strSQL, adoCon

'Delete the record from the database

rsDeleteEntry.Delete

'Reset server objects

rsDeleteEntry.Close

Set rsDeleteEntry = Nothing

Set adoCon = Nothing

'Return to the delete select page incase another record needs deleting

Response.Redirect "delete\_select.asp"

%>