Webster UNIVERSITY Vienna

Accessing Databases

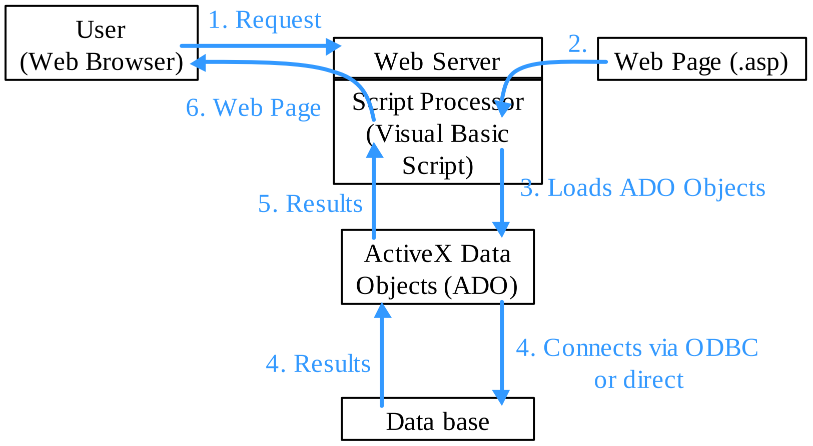
Michael Hahsler

22.2.2002

Contents

- Processing of ASP
- ADO Objects
  - Connection
  - Recordset
  - Record
  - Command
  - Error

Processing of Active Server Pages



User (Web Browser)

1. Request

Web Server

2.

Web Page (.asp)

Script Processor (Visual Basic Script)

6. Web Page

3. Loads ADO Objects

5. Results

ActiveX Data Objects (ADO)

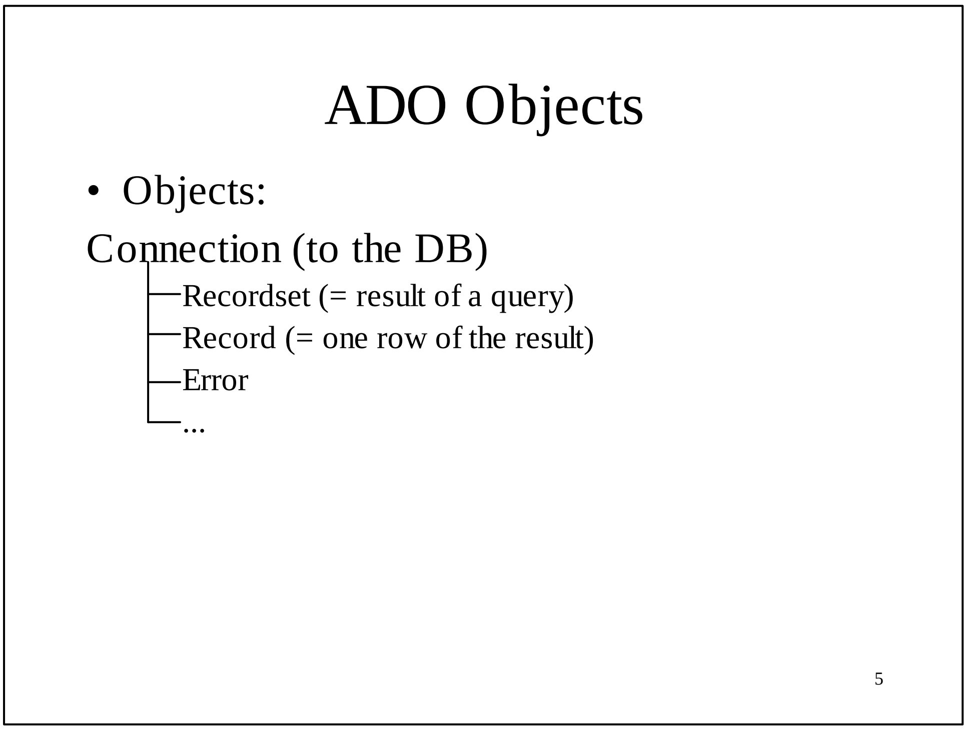
4. Results

4. Connects via ODBC or direct

Data base

How to get ADO

- ADO is part of MDAC (MS Data Access Components) and comes with various MS products:
  - Windows
  - IIS-Internet Information Services (Web-Server)
  - MS Internet Explorer...
- Look at: http://www.microsoft.com/data/ for info about ADO, ODBC,... with MS Windows

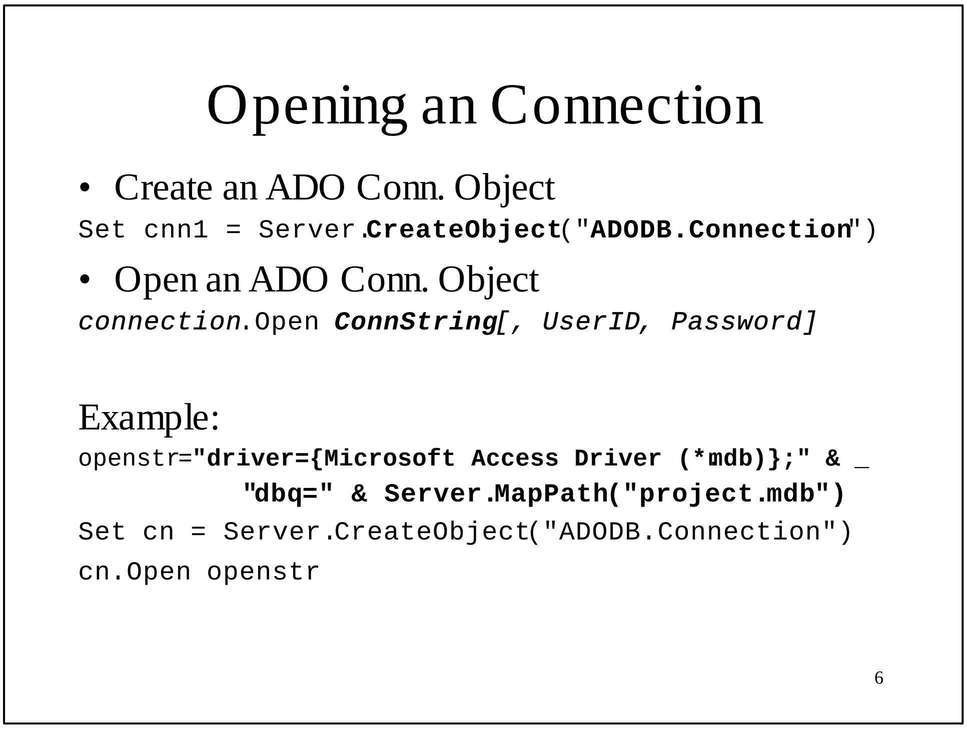


ADO Objects

- Objects:

Connection (to the DB)

```
├─Recordset (= result of a query)
├─Record (= one row of the result)
├─Error
└─...
```



Opening an Connection

- Create an ADO Conn. Object

```
Set cnn1 = Server.CreateObject("ADODB.Connection")
```

- Open an ADO Conn. Object

```
connection.Open ConnString[, UserID, Password]
```

Example:

```
openstr="driver={Microsoft Access Driver (*mdb)};" & _
          "dbq=" & Server.MapPath("project.mdb")
Set cn = Server.CreateObject("ADODB.Connection")
cn.Open openstr
```

Closing an Connection

- Close

```
cnn1.Close
```

- Release the object memory

```
Set cnn1 = Nothing
```

ODBC Data Sources

- ODBC DBs are identified by a **DSN** (Data Source Name) set by the Operating System

- Define a new DSN

Control Panel -> Administrative Tools -> ODBC

Opening an ODBC Connection

```
Set cnn1 = Server.CreateObject("ADODB.Connection")

cString = "Provider=ODBC; DSN=grades"
cnn1.Open cString

.....

cnn1.Close
Set cnn1 = Nothing
```

Accessing Tables

- The most important Object is the Recordset
- It has many methods to:
  - Obtaining results from a DB
  - Move within the result (records)
  - Update (Batch) ...
- And even more properties:
  - Cursor type (move within a recordset)
  - EOF, Index ...

Creating a Recordset

```
Set rs = Server.CreateObject("ADODB.Recordset")
rs.Open Source, ActiveConnection,
  CursorType, LockType[, Options]
```

Example:

```
sql="SELECT * FROM members"
rs.Open sql, conn1, 0, 1
```

(see book pp .71-73)

Close: `rs.Close    set rs = Nothing`

Move through a Recordset

```
rs.MoveFirst      rs.MoveLast
rs.MoveNext       rs.MovePrev
rs.EOF            rs.BOF
```

Example:

```
rb.MoveFirst
Do While Not rs.EOF
    ' Code
  rb.MoveNext
Loop
```

Move through a Recordset II

```
rs.Move 10[, Startpoint]
```

Startpoint can be:

- 0 ... Current position
- 1 ... First record
- 2 ... Last record

Problem: Positions are not stable (deletion, filtering)

-> Bookmarks

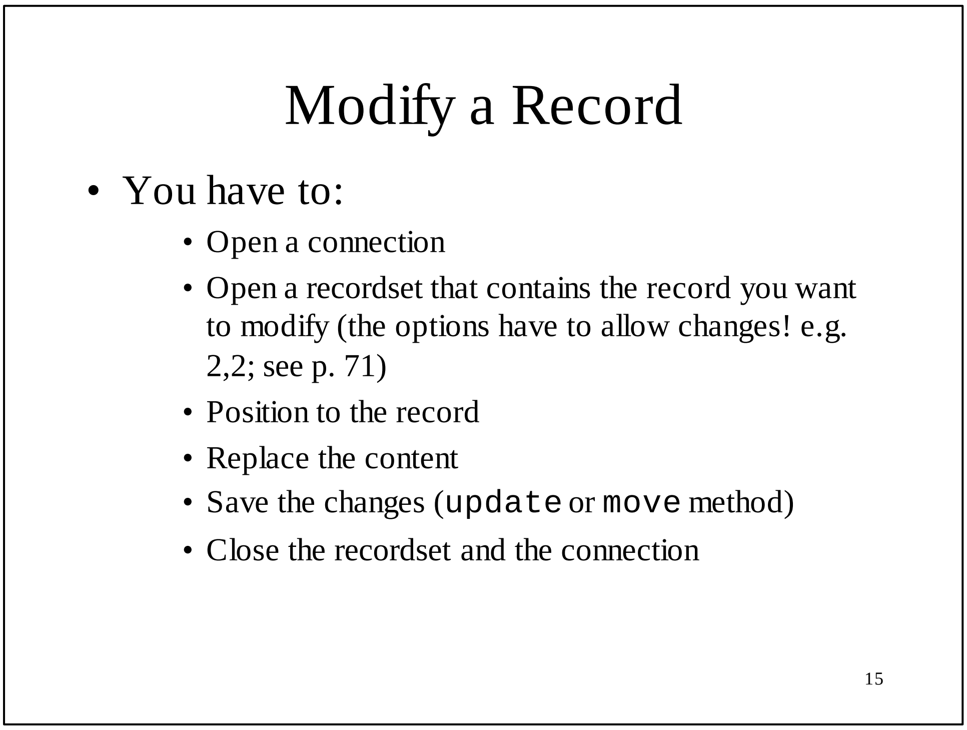
Access Fields in a Recordset

- The record has to be selected by the move method
- Then you can get the fields of the selected record

`rs.Fields("memberid").Value` or
`rs.Fields(0).Value` or
`rs("memberid")` or `rs(0)`

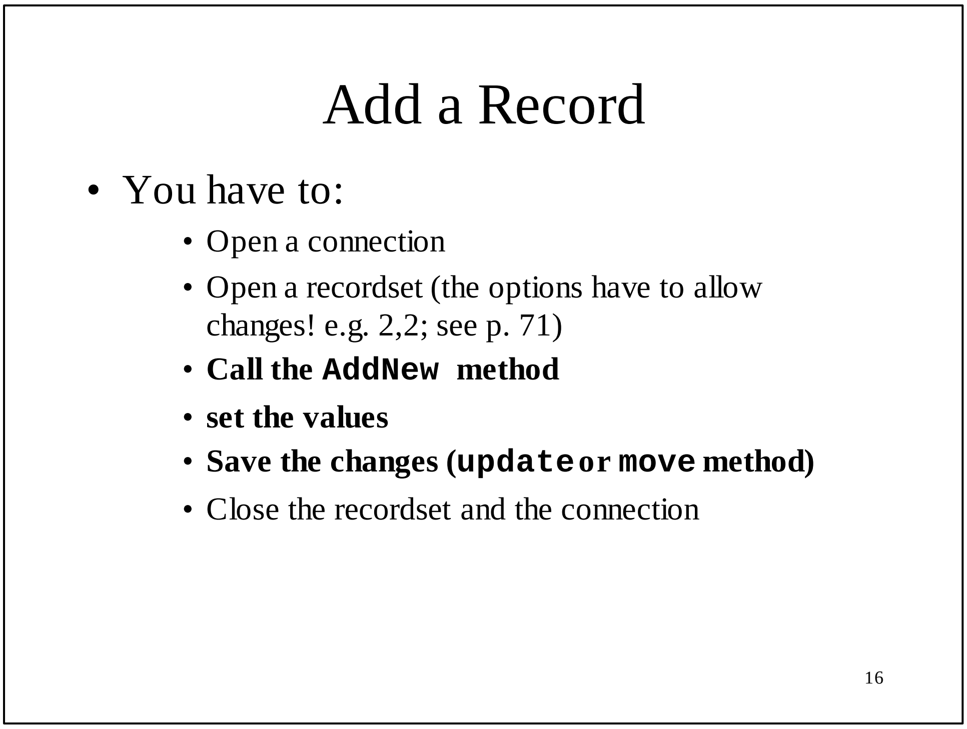
Example:

```
For i=0 To rs.Fields.Count-1
Response.Write "<td>" & rs.Fields(i) &_
  "</td>"
Next
```



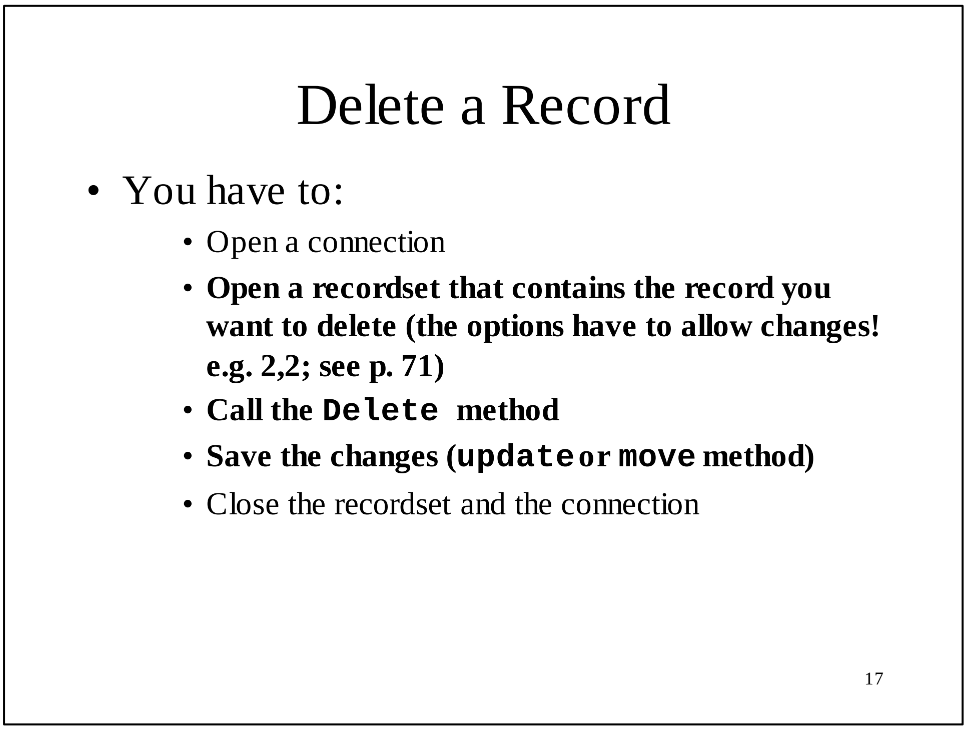
Modify a Record

- You have to:
  - Open a connection
  - Open a recordset that contains the record you want to modify (the options have to allow changes! e.g. 2,2; see p. 71)
  - Position to the record
  - Replace the content
  - Save the changes (`update` or `move` method)
  - Close the recordset and the connection



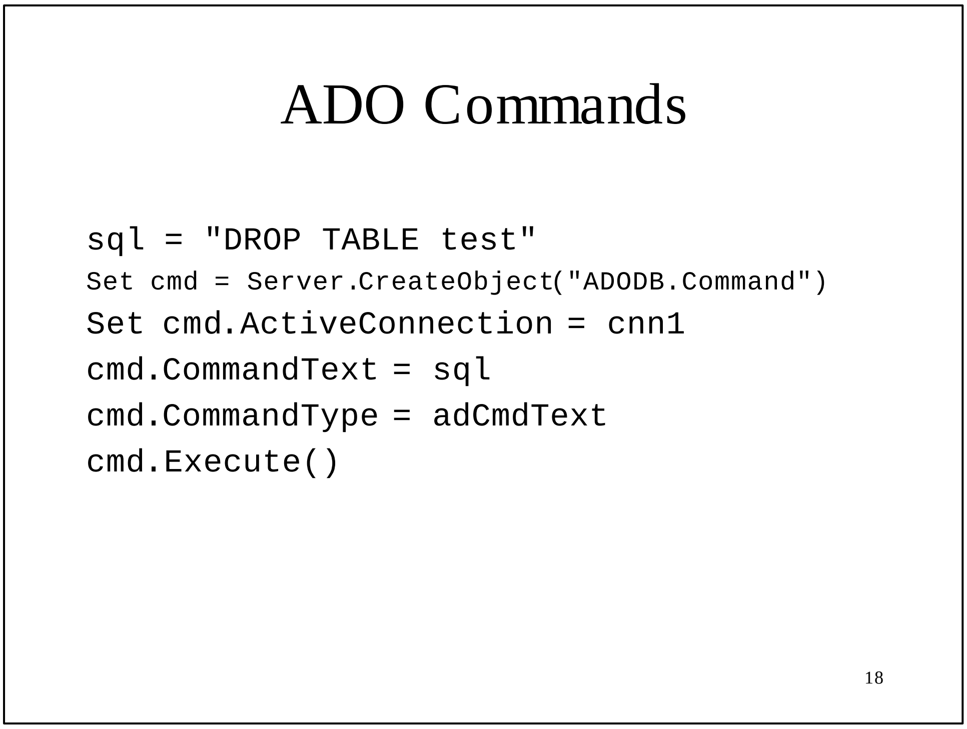
Add a Record

- You have to:
  - Open a connection
  - Open a recordset (the options have to allow changes! e.g. 2,2; see p. 71)
  - **Call the `AddNew` method**
  - **set the values**
  - **Save the changes (`update` or `move` method)**
  - Close the recordset and the connection



Delete a Record

- You have to:
  - Open a connection
  - **Open a recordset that contains the record you want to delete (the options have to allow changes! e.g. 2,2; see p. 71)**
  - **Call the `Delete` method**
  - **Save the changes (`update` or `move` method)**
  - Close the recordset and the connection



ADO Commands

```
sql = "DROP TABLE test"
Set cmd = Server.CreateObject("ADODB.Command")
Set cmd.ActiveConnection = cnn1
cmd.CommandText = sql
cmd.CommandType = adCmdText
cmd.Execute()
```

Stored Query (ADO Parameter)

- Queries can be predefined in the DB (stored procedure)
- **Advantage:** No SQL Code in the script
- see Book pp. 97-100

ADO Error

- Program goes on after error

```
On Error Resume Next
```

Example:

```
On Error Resume Next
...
cnn1.Open ConnectionStr, "", ""
if Err Then
  Response.Write "<p>Cannot connect: " &_
     Err.Number &"-"& Err.Description &_
  "</p>"
End If
On Error GoTo 0
```