

Az ADO használata

Az ActiveX Data Objects adatbázisokkal dolgozik. Minden adatbázissal kapcsolatos feladat kivitelezhető vele. Ezek közül a könyvben csak az adatbázis megnyitásával, a lekérdezés futtatásával és a rekordok írásával, olvasásával fogunk foglalkozni. Korai kötéshez a Microsoft ActiveX Data Objects Library-t kell bepipálni.

Adatbázis megnyitása

Egy adatbázist a következő módon nyithatunk meg.

```
'Korai kötés: Dim oConn As New ADODB.Connection
Set oConn = CreateObject("ADODB.Connection")
oConn.Open "File Name=adatok.udl"
```

A kódban az adatok.udl fájl adja meg az adatbázis-kapcsolatot. A fájlt a programozónak kell létrehozni, ehhez hozzunk létre egy új szöveges dokumentumot adatok.udl néven. A fájlra duplán kattintva az Adatkapcsolat tulajdonságai ablak jelenik meg. Access adatbázis megnyitásához válasszuk ki a Jet Providert.



A szolgáltató kiválasztása után a *Kapcsolat* fülön adhatjuk meg az elérési utat, és le is ellenőrizhetjük a *Kapcsolat* tesztelése gombbal.

A *Speciális* fülön az engedélyeket állíthatjuk be. Ha csak olvasni akarunk, biztonságos a *Read* beállítás, ha írni is akarunk, a *ReadWrite* szükséges. *Share Exclusive* esetén más nem nyithatja meg egyidejűleg az adatbázist, *Share Deny Read/Write* esetén az írást, olvasást tiltjuk meg a többieknek. Ezek a korlátozások az egész adatbázisra vonatkoznak. Érdemesebb a többi felhasználót is beengedni a *Share Deny None* pipával, és a lekérdezéskor letiltani a rekordokhoz a hozzáférést.

Ha az udl-t jegyzetömbben nyitjuk meg, a következő három sort láthatjuk:

```
[oledb]
; Everything after this line is an OLE DB initstring
Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=users.mdb;Persist Security Info=False
```

Az utolsó sort használhatjuk az adatbázis megnyitásakor a fájlhivatkozás helyett:

```
oConn.Open " Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=users.mdb;Persist Security Info=False "
```

Lekérdezés az adatbázisból

Miután kész az adatkapcsolat objektum (az előző példában oConn), hozzunk létre egy recordset objektumot:

```
'Korai kötés: Dim oRS As New ADODB.Recordset
Set oRS = WScript.CreateObject("ADODB.Recordset")
```

Az objektum következő tulajdonságaira és eljárásaira lesz szükségünk. (Ezek csak a legfontosabbak, töredéke a tulajdonságoknak és metódusoknak; ahol lehetett, az opcionális paraméterek is kimaradtak.)

A recordset tulajdonságai

.BOF: **Igaz-Hamis:** igaz, ha az első előtti rekord van kijelölve.

.EOF: **Igaz-Hamis:** igaz, ha az utolsó utáni rekord van kijelölve. Ha az EOF vagy BOF igaz, akkor az olvasás hibát okoz.

.Fields(*index*): Ez az alapértelmezett tulajdonság. Leegyszerűsítve kezelhetjük úgy, mint egy gyűjteményt, ami az épp kijelölt rekord tulajdonságait tartalmazza. Az *index* lehet szám vagy tulajdonságnév. Mivel az adatbázis változhat, a tulajdonságnevet használjuk, így a kód is olvashatóbb lesz.

A Field (adatbázis tulajdonság) objektum alapértelmezett tulajdonsága a Value (érték). Ezen kívül a Name tulajdonságra lehet szükségünk, ami a Field adatbázisbeli nevét tartalmazza.

A recordset metódusai

.Open *lekérdezés, kapcsolatobjektum, [kurzortípus], [zárolástípus], [opciók]*

A *lekérdezés* egy SQL parancs. A *kapcsolatobjektum* egy aktív ADODB.Connection objektum.

A *kurzortípus* jelentését a következő táblázat foglalja össze:

Konstans neve	Értéke	Jelentése
adOpenForwardOnly	0	Alapértelmezett. Helyi másolat a rekordokról. A leggyorsabb, a kurzor csak előrefele léphet.
adOpenStatic	3	Helyi másolat a rekordokról. A kurzor bármerre léphet.
adOpenKeyset	1	Látható, ha időközben más változtatta a rekordot, de a hozzáadás, törlés nem látható
adOpenDynamic	2	Az időközben hozzáadott és törölt rekordok is látszanak. Nagy forgalmat generál.

A zárolástípus lehetséges értékei és azok jelentése:

Konstans neve	Értéke	Jelentése
adLockReadOnly	1	Az általunk megnyitott rekordok nem írhatók. (Alapértelmezett beállítás.)
adLockPessimistic	2	A rendszer mindent elkövet, hogy elkerülje a közös használatból adódó problémákat.
adLockOptimistic	3	A rendszer nem készül fel arra, hogy több felhasználó egyszerre használja a rekordokat.
adLockBatchOptimistic	4	Az adLockOptimistic-hez hasonlóan jár el, és az adatok visszairása nagyobb kötegekben történik.

Végül az *opciók* egy szám, ami a lekérdezés értelmezését pontosítja. Egyszerű lekérdezések (select, delete stb.) esetén 1 az értéke.

`.Close`: Feltölti az aktuális rekordot és bezárja a RecordSetet. Csak ezután lehet újra megnyitni.

`.Update`: Rekord feltöltése, módosítása. A kurzor mozgásánál, bezárásnál automatikusan megtörténik, tehát általában nincs rá szükség.

`.UpdateBatch`: Kötegelt feltöltés.

`.CancelUpdate`: Feltöltés mellőzése. Ha az utolsó feltöltés nem sikerült valamilyen kritérium miatt, akkor csak a CancelUpdate után lehet bezárni a RecordSetet.

`.AddNew`: Új rekordot szúr be, és ráállítja a kurzort.

`.Delete`: A kijelölt rekordot törli.

`.MoveFirst`: Az első rekordot jelöli ki.

.MoveLast: Az utolsó rekordot jelöli ki, adOpenForwardOnly kurzor esetén hibát okoz.

.MoveNext: A következő rekordot jelöli ki.

.MovePrevious: Az előző rekordot jelöli ki, adOpenForwardOnly kurzor esetén hibát okoz.

Példa adatbáziskezelésre

Létrehoztunk egy data.udl-t, ami az Office 97-ben található példa-adatbázisra, a Házi leltár.mdb-re mutat. Bármilyen más adatbázis is jó, de akkor a Select utasításban át kell írni a tábla nevét.

A szkript vagy makró beolvassa a tábla egész tartalmát, és egy html táblázatba (lista.htm) írja, oszlopfejlécekkel együtt. (A html dokumentumok felépítésével az utolsó fejezet foglalkozik.)

```
Set oConn = CreateObject("ADODB.Connection")
oConn.Open "File Name=data.udl"

Set oRS = CreateObject("ADODB.Recordset")
oRS.Open "Select * From [Házi leltár]", oConn,,,1

Set oFSO = CreateObject("Scripting.FileSystemObject")
Set f = oFSO.CreateTextFile("lista.htm")

'Táblázat beszúrása
f.WriteLine "<table border='1'>"
f.WriteLine "<TR>"      'Első sor
For Each oField In oRS.Fields
    'Oszlopfejlécek + nem törhető szóköz
    f.WriteLine " <TH>" & oField.Name & " &nbsp;</TH>"
Next

f.WriteLine "</TR>"      'Első sor lezárása

Do Until oRS.EOF
    f.WriteLine "<TR>"      'Új sor

    For Each oField In oRS.Fields
        'Cella + nem törhető szóköz
        f.WriteLine " <TD>" & oRS(oField.Name) & " &nbsp;</TD>"
    Next
```

```
f.WriteLine "</TR>"      'Sor lezárása
oRS.MoveNext
Loop

f.WriteLine "</table>"   'Táblázat lezárása

oConn.Close
f.Close
```

Hibakezelés

Az adatbázis hibák alapesetben megszakítják a szkriptet. Ha kikapcsoljuk a beépített hibakezelést (`On Error Resume Next`), akkor a `Connection` objektum `Errors` gyűjteményben (pl.: `oConn.Errors`) kell megvizsgálnunk az adatbázis hibaüzeneteit. A gyűjtemény tagjai `Error` objektumok, melyeknek alapértelmezett tulajdonsága a `Description`, azaz leírás. Ezen kívül hasznos lehet a `Number` tulajdonság, ami a hiba kódszámát tárolja.

A hibákat kitörölni a következő paranccsal lehet:

```
oConn.Errors.Clear
```

Feladatok

1. Állítsunk be és használjunk olyan adatkapcsolatot, mely Excel munkalapról, különféle szöveges dokumentumokból kérdez le!

2. Hozzunk létre olyan adatbázist, mely fájlok nyilvántartására való (elérési út, méret, típus, létrehozás dátuma)! Írjunk olyan szkriptet, mely az adatbázishoz hozzáadja a ráhúzott fájlokat (drag'n'drop)!