

Microsoft®

100% technológia • 0% marketing

TechNet

# Az SQL Server 2005 újdonságai

Az ADAM és a .NET  
Active Directory – webes regisztráció  
A Windows szoftveres RAID megoldásai  
Client Callback az ASP.NET-ben



VI./4. szám  
2005. augusztus



# Legyen Ön is a NAGYok között!

## Ősztől ismét Microsoft és Cisco mérnök képzések

**Microsoft**  
GOLD CERTIFIED  
Partner

Learning Solutions

Hivatalos Microsoft valamint Cisco tanfolyamokra épülő, rugalmas beosztású, kedvezményes konstrukciójú és árú képzéssorozatok. Különböző segédanyagokat, vizsgafelkészítő anyagokat tartalmazó oktatási konstrukciók és csomagok. Kedvezményes lehetőségek további tanfolyamokra, vizsgákra és felkészítő anyagokra.

### Válassza ki az Önnek megfelelő minősítést és képzési konstrukciót!

#### **Microsoft Windows 2003 rendszeradminisztrátor (MCSA) képzés**

Két tanfolyam (80 óra) – 279.000 Ft-tól

Kezdés: szeptember 26.

#### **Microsoft Windows 2003 rendszermérnök (MCSE) képzés**

Négy tanfolyam (160 óra) – 559.000 Ft-tól

Kezdés: szeptember 26.

#### **Microsoft SQL 2000/2005 adatbázis-adminisztrátor (MCDBA) képzés**

Négy tanfolyam (144 óra) – 509.000 Ft-tól

Kezdés: szeptember 26.

#### **Microsoft .NET fejlesztői (MCAD) képzés**

Négy tanfolyam (184 óra) – 669.000 Ft-tól

Kezdés: október 17.

#### **Cisco hálózati rendszermérnök képzés (CCNA/CCNP)**

Négy tanfolyam (136 óra) – 666.000 Ft-tól

Kezdés: október 3.

**A tanfolyamokra beváltható a Microsoft frissítési garancia  
tanfolyambón (SA voucher) is.**

**Valamennyi tanfolyam szakképzési hozzájárulásból elszámolható!**

A tanfolyamokkal és akciókkal kapcsolatos további tudnivalók weboldalunkon találhatóak, vagy kérjük, keresse szervezőnket! A feltüntetett árak a 25% áfát nem tartalmazzák.

**SZÁMALK Továbbképzés, 1115 Budapest, Etele út 68.**

**Simon Ferenc, 203-0304/4122 mellék, simonf@szamalk.hu**

**[www.szamalk.hu/tisza](http://www.szamalk.hu/tisza)**

**[www.szamalk.hu/tisza](http://www.szamalk.hu/tisza) (Akciók)**



*Partner az oktatási megoldásokban*



**TechNet Magazin**

VI. évfolyam, 4. szám  
2005. augusztus

Szerkesztőség és kiadó:

**Microsoft Magyarország Kft.**

1031 Budapest, Graphisoft park 3.

Feladás kiadó:

**Szekely Tamás** marketingigazgató

Szerkesztő:

**Takács Gitta** (Epsilon Press)

Szaklektor:

**Fóti Marcell** (Netacademia)

Lapterv és nyomdai előkészítés:

**Dobák Ildikó**  
(Ars Luna Bt.)

Borítófóto:

**Archív**

Nyomda:

**AduPrint Kiadó és Nyomda Kft.**

1033 Budapest, Csikós utca 8.

Feladás vezető: Tóth Béláné

Webcím:

[www.microsoft.com/hun/technet/](http://www.microsoft.com/hun/technet/)

E-mail:

[technetmagazin@microsoft.hu](mailto:technetmagazin@microsoft.hu)

ISSN 1586-5185

A TechNet Magazinban közölt cikkek, képek és illusztrációk csak a kiadóval történt előzetes egyeztetés után használhatók fel.

Adatvédelmi tájékoztató: Az Ön adatai a Microsoft Magyarország adatbázisából származnak. Amennyiben nem kívánja, hogy a továbbiakban a TechNet Magazinban vagy más ajánlatokkal keressük meg Önt, bármikor kérheti adatainak törlését a Microsoft Magyarország Kft. címéről írott levélben vagy e-mailben.

# Microsoft Office 12 alapértelmezett XML fájlformátummal

**A MICROSOFT OFFICE KÖVETKEZŐ VERZIÓJA – AMELY AZ OFFICE 12 KÓDNEVET VISELI – A SZABVÁNYNAK SZÁMÍTÓ XML TECHNOLÓGIÁT FOGJA ALAPÉRTELMEZETT FÁJLFORMÁTUMKÉNT HASZNÁLNI.**

**A** Microsoft Office Open XML Formats nevű új fájlformátum a várhatóan 2006 második felében piacra kerülő Microsoft Office Word, Excel és PowerPoint Office 12 verzióinak lesz az alapértelmezett fájlformátuma. A Microsoft az Office 2000 óta támogatja ezt az iparági szabványt, ugyanis ekkor vezette be a cég az XML alapú dokumentumtulajdonságokat. Az Office XP, majd az Office 2003 megjelenésével fokozatosan tovább szélesítette ezt a támogatást.

Jelenleg több mint egymillió fejlesztő készített megoldásokat az Office 2003 használatával, és harmaduknál is többen XML-t használnak megoldásaikban. A Gartner Group becslése szerint az XML használatára alkalmas elektronikus űrlapok alkalmazása legalább kétszeresére fog nőni a következő évben, 2007-re pedig az infómunkások 40 százaléka fog az XML használatára felkészített eszközöket használni a tartalom létrehozására. A Forrester előrejelzései azt mutatják, hogy 2008-ra az XML lesz az adatok archiválására szolgáló meghatározó dokumentumformátumok egyike.

Az Open XML Formats olyan robusztus, tömör fájlformátum, amely kisebb fájlméretet, a hiányos vagy sérült fájlok továbbfejlesztett helyreállítását és nagyobb biztonságot kínál. Mivel az új formátum fájlok a megfelelő Microsoft Office 2003 fájlokkal összehasonlítva akár 75 százalékkal kisebb méretűek, az ügyfelek adattárolási igényei és költségei jelentős mértékben csökkenthetők. A kisebb fájlok kevesebb helyet foglalnak el az elektronikus levelek mellékleteiben és a letölthető fájlok esetében is, így a sávszélességgel kapcsolatos költségek csökkentésére is számítá-

ni lehet. A kisebb fájlméretet az iparági szabványnak számító ZIP technológia teszi lehetővé, amely automatikusan tömöríti a fájlokban található összetevőket, majd magát a fájlt is.

Az új fájlformátum fejlett adathelyreállító képességei lehetővé teszik a fájl sérteletlen részeinek megnyitását és használatát, amikor csak a fájl egyik összetevője – például egy grafikon vagy egy képsérült. A Microsoft Office Open XML Formats ezen felül olyan technológiát is tartalmaz, amely a fájlok megnyitásokor észleli a sérüléseket, és megkísérli azok helyreállítását. Ezek a lehetőségek a formátum struktúrájára épülnek, amely különálló összetevőkké bontja a fájlok adatainak tárolását, amelyeket egymástól függetlenül lehet beolvasni, karbantartani és kezelni.

A Microsoft Office Open XML Formats nyílt fájlformátum, amelyhez jogdíjmentes licenc tartozik. Bárki közvetlenül beépítheti kiszolgálóba, alkalmazásai- ba és üzleti folyamataiba.

A nyílt, jogdíjmentes licenc segít abban, hogy a külső fejlesztők egyszerűen integrálhassák a fájlformátumot eszközeikkel, és olyan megoldásokat hozzanak létre, amelyek általános hozzáférést nyújtanak a Microsoft Office alapú adatokhoz, anélkül, hogy szükség lenne a Microsoft Office alkalmazásokra vagy a létrehozáshoz használt eszközökre.

## **További info:**

[www.microsoft.com/office/preview](http://www.microsoft.com/office/preview)

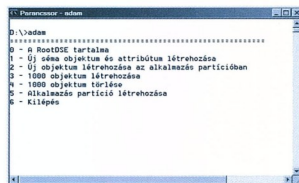
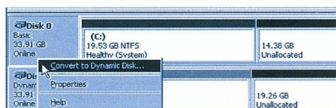
Aki ezen a weboldalon regisztrál, folyamatosan hozzájuthat az Office új változatával kapcsolatos legfrissebb bejelentésekhez, valamint a későbbiekben az 'Office 12' Beta 2 verziójához.



## Az ADAM címtár használata II.

### ADAM ÉS A .NET

Cikkünk második részében további megpróbáltatásoknak tesszük ki az előző alkalommal telepített ADAM példányunkat. Ezúttal azonban már nem az ADAM csomagban érkező felügyeleti eszközöket fogjuk használni, hanem a .NET keretrendszer DirectoryServices névterének osztályait.



## RAID-ers

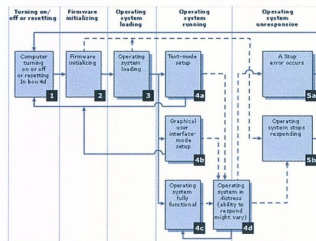
### A DROMEDÁR ESETE A SÍVATAGBAN A RAID-EREKKEL 2. RÉSZ

A kiszolgálókban mindennapos a RAID alkalmazása, gyakran azonban a használói sincsenek egészen tisztában azzal, hogy pontosan mit, illetve miért használnak. Cikkemben összefoglalom mindazt, amit a RAID-ról tudni érdemes. Ebben a részben a Windows szoftveres RAID megoldásait mutatom be. Bár én a magyar nyelvű kifejezéseket használom, a példákat angol nyelvű operációs rendszer segítségével mutatom be, mert a kiszolgálókra leggyakrabban angol nyelvű operációs rendszert telepítenek.

## Windows szolgáltatók 8.rész

### TOVÁBBRA IS A LOCALSYSTEM FIÓK HASZNÁLATÁRÓL

Már csak két kupac szolgáltatás maradt ismertetésre, ám ezek a lista végén szereplők sem kevésbé fontosak/érdekesekek, mint az eddigiék.



## VPN karantén 2. rész

### WINDOWS SERVER 2003 + ISA SERVER 2004 KÖRNYEZETBEN • A CONNECTION MANAGER ADMINISTRATION KIT SZEREPE

Cikksorozatunk előző részében eljutottunk a bevezetésen, a működés elméleti összefoglalásán keresztül az előkészítésig, azaz a Resource Kit passzoló elemeinek telepítéséig illetve a regisztrációs adatbázisban történő beállításáig. Az ISA 2004-en elkészítettük a speciális protokollt és szabályt, a végén engedélyeztük magát a karantént is. Ebben a részben megismerkedünk a Connection Manager Administration Kit-tel [amely nem csak VPN-Q eszköz, más távélérésű kapcsolatoknál is remekül használható], és elkészítjük a Connection Manager profilt, amelyben a VPN-Q kliens is benne lesz.



# ASP.NET 2.0 (Whidbey)

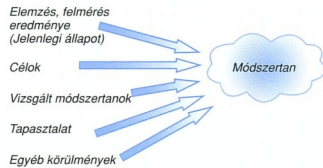
## Mi várható a 2005. évi ASP.NET-ben?

CLIENT CALLBACK (ÜGYFÉLOLDALI VISSZAHÍVÁS)

Kevésbé reklámozott, de nagyon hasznos és trükkös új szolgáltatás a Client Callback. Segítségével anélkül változtathatjuk meg a böngészőbe betöltött lap tartalmát, hogy a hagyományos postázásos módszerrel legeneráltatnánk újra a teljes lapot, így sokkal gyorsabb válaszidőket produkáló webalkalmazásokat írhatunk.

## IT folyamatok javítása

Előző cikkemben egy, a hazai IT cégek körében végzett felmérést ismertettem. Most ahhoz kapcsolódóan mutatok be néhány, a folyamatok javítására irányuló ötletet, praktikát.



## Ami a hivatalos Microsoft tanfolyamokból kimaradt...

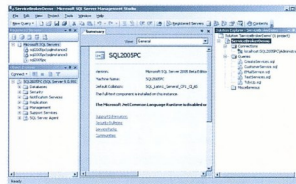
ACTIVE DIRECTORY - WEBES REGISZTRÁCIÓ (2. RÉSZ)

Folytatjuk weblapunk készítését, melynek segítségével a felhasználók maguk kérvényezhetik hozzáférési szándékukat a rendszerünkhöz, intranetünkhöz. A már elkészített önregisztráció alapján, az automatikusan létrehozott felhasználói fiókokat tovább boncoljuk, és elkészítjük az e-mail alapú értesítéseket, továbbá megnézzük, hogyan „élesedik”, és vándorol a létrehozott felhasználói fiók egyik OU-ból a másikba. De először megnézzük, miként lehet a „shell”-t IE-re cserélni az önregisztráció első lépésőjéhez.

## Dr. Watson

Az SQL 2005 ÚJDONSÁGAI I. RÉSZ

Jön! Jön! Jön! Csak még azt nem tudjuk, mikor! De előbb-utóbb megújul az immár öt éves adatbáziskezelő, az SQL Server. E cikkben összeválogattam egy-két olyan újonságot, ami messze túlmutat a kozmetikai változtatásokon. Ezek közül több akár egy teljes cikket megérdemelne, pedig a pusztas felsorolásuk is két cikkbe fér csak bele. Kozmetikai újonság is van persze: az sp\_addlogint felváltja a CREATE LOGIN, az sp\_addusert felváltja a CREATE USER, és általában minden parancs SQL-esekben fog kinézni. De lássuk a lényegét!



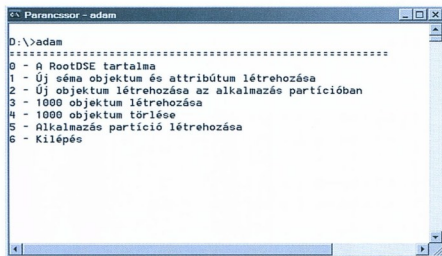


# Az ADAM címtár használata II.

## ADAM ÉS A .NET

Cikkünk második részében további megpróbáltatásoknak tesszük ki az előző alkalommal telepített ADAM példányunkat. Ezúttal azonban már nem az ADAM csomagban érkező felügyeleti eszközöket fogjuk használni, hanem a .NET keretrendszer DirectoryServices névterének osztályait.

Újabb sémabővítést fogunk végezni (objektum típusát és attribútumot is létrehozunk), majd objektumokat gyártunk az új séma elemek alapján. A programozott hozzáférés lehetőségeit kihasználva megpróbálkozunk objektumok tömeges létrehozásával és törlésével is, hogy megfigyelhessük milyen sebességre képes az „objektumgyár”. Az említett műveleteket a cikkhez tartozó mintaprogram (letölthető az [1] címről) segítségével próbálhatjuk ki. A program konzol alapú, menüszerkezete a következő lesz:



```
Parancsok - adam
D:\>adam
=====
0 - A RootDSE tartalma
1 - Új séma objektum és attribútum létrehozása
2 - Új objektum létrehozása az alkalmazás partícióban
3 - 1000 objektum létrehozása
4 - 1000 objektum törlése
5 - Alkalmazás partíció létrehozása
6 - Kilépés
```

A mintaprogram menüszerkezete

### Előkészületek

Az ADAM példány telepítését és kezdeti beállításainak megadását a cikk előző része alapján végezhetjük el. Célserű egy frissen telepített példánnyal kísérletezni, hogy ne kelljen a korábbi változtatások miatt előálló esetleges ütközésekre figyelniük. Telepítés közben mindenképpen szükséges az „MS-User.Idf” nevű fájl importálása. Ha esetleg telepítés közben meglepedeznének róla, később az Ildife program használatával pótolhatjuk ezt a hiányosságot.

### A RootDSE

Hogy programunk hozzáférhessen a címtár séma partíciójához, tudnunk kell annak DN-jét, vagyis megkülönböztető nevét, amelynek formája:

```
CN=Schema, CN=Configuration, CN={GUID}
```

A névben szereplő GUID értéket természetesen nem tudhatjuk akkor, amikor programunk készül. A DN-t a címtár RootDSE objektumának tulajdonságai között kell megkeresnünk. A címtárhoz csatlakozni kívánó programok ezt az információt bejelentkezés nélkül is elolvashatják.

A következő programrészlet a RootDSE névtérhez csatlakozó DirectoryEntry objektumot hoz létre, és kilistázza az objektum valamennyi tulajdonságának nevét, és értékeit (a tulajdonságok egy részéhez több érték is tartozik).

```
static void ListRootDSE(){
    string strHostName = GetName("Kérem az ADAM
    ❖ kiszolgáló nevét (alapértelmezés: localhost): ",
    ❖ "localhost");
    string strPortNumber = GetName("Kérem a
    ❖ portszámot (Alapértelmezés: 389): ", "389");
    DirectoryEntry objRoot = new
    ❖ DirectoryEntry("LDAP://" + strHostName + "/" +
    ❖ strPortNumber + "/RootDSE");
    foreach(string propertyName in
    ❖ objRoot.Properties.PropertyNames) {
    ❖ //egyes tulajdonságok több értéket is
    ❖ tartalmaznak
    ❖ int count =
    ❖ objRoot.Properties[propertyName].Count;
    ❖ Console.WriteLine(propertyName + ": ");
    ❖ //végiglépkedünk az adott tulajdonság összes
    ❖ értéken
    ❖ for( int i = 0; i < count; i++)
    ❖ Console.WriteLine(objRoot.Properties[
    ❖ propertyName][i].ToString());
    ❖ } //foreach
}
```

A DirectoryEntry objektum konstruktorában láthatjuk, hogy ADAM címtárhoz való csatlakozáskor nem elég a kiszolgáló nevének megadása (mint Active Directory esetében), meg kell határozniunk a csatlakozáshoz használandó port számát



is. Egy számítógépen több ADAM példány is futhat, így csak ezek az adatok azonosítják egyértelműen a megcélzott ADAM példányt.

Ha végignézzük a kapott listát, megtalálhatjuk a „schemaNamingContext” sort. Ennek értéke például:

```
CN=Schema,CN=Configuration,CN={4C099541-ABCF-4BFF-  
%8DOB-2F16734A353C}
```

Nem lenne túl kellemes, ha ezt kellene beleírniuk a forráskódba...

## Attribútum létrehozása

Miután sikeresen lekérdeztük a séma partíció DN-jét, létrehozhatjuk a hozzá csatlakozó DirectoryEntry objektumot, ennek segítségével hozzhatjuk létre az új attribútumot. Az alábbi kódrészletben először csatlakozunk a megfelelő ADAM példányhoz, majd lekérdezzük a RootDSE objektum „schemaNamingContext” tulajdonságát, és csatlakozunk az ADAM példány séma partíciójához.

```
//Csatlakozunk az ADAM példány gyökerére  
string strHostName = GetName("Kérem az ADAM  
%kiszolgáló nevét (alapértelmezés: localhost): ",  
% "localhost");  
string strPortNumber = GetName("Kérem a portszámot  
%(Alapértelmezés: 389): ", "389");  
DirectoryEntry objRoot = new  
%DirectoryEntry("LDAP://"+strHostName+": "+  
%strPortNumber+"/RootDSE");  
//lekérjük a séma partíció DN-jét  
string strSchemaNamingContext =  
%objRoot.Properties["schemaNamingContext"].Value.  
%ToString();  
//létrehozuk a séma partícióra mutató  
%DirectoryEntry objektumot  
DirectoryEntry objSchema = new  
%DirectoryEntry("LDAP://"+strHostName+ " "+  
%strPortNumber+ "/" +strSchemaNamingContext);  
Console.WriteLine("A séma partíció útvonala:  
%{0}",objSchema.Path);
```

Természetesen az attribútum maga is egy séma objektum (amelynek osztálya „attributeSchema”), így neki magának is vannak attribútumai, ezek közül néhányat már a létrehozás során be kell állítanunk:

```
DirectoryEntry objNewAttribute;  
const string strNewAttributeName = "weblap";  
string strCNNewAttributeName = String.Concat("CN=",  
%strNewAttributeName);  
//beállítjuk az attribútum kötelező tulajdonságait  
objNewAttribute = objSchema.Children.  
%Add(strCNNewAttributeName, "attributeSchema");  
objNewAttribute.Properties["oMSyntax"].Add(64);  
objNewAttribute.Properties["LDAPDisplayName"].  
%Add(strNewAttributeName);  
objNewAttribute.Properties["isSingleValued"].  
%Add(true);  
objNewAttribute.Properties["attributeSyntax"].  
%Add("2.5.12");  
objNewAttribute.Properties["attributeID"].  
%Add("1.2.840.113556.1.4.296");  
objNewAttribute.CommitChanges();  
Console.WriteLine("A létrejött attribútum: " +  
%objNewAttribute.Name);
```

A kódrészletből látható, hogy az új attribútumot (objektumot) a séma partícióra mutató DirectoryEntry objektum „Children” kollekciójához kell hozzáadnunk. Az „Add” metódus egy másik DirectoryEntry-t ad vissza, ennek tulajdonságait kell meghatározniuk.

Az egyes tulajdonságok jelentése a cikk előző részében megtalálható, így erre most nem térünk ki.

## Objektum típus létrehozása

Új objektum típus létrehozásához természetesen szintén szükségünk van a séma partíció DN-jére. Ezúttal a Children kollekcióhoz „classSchema” osztályú objektumot kell hozzáadnunk. A kötelező attribútumok mellett itt adhatjuk meg az új típus saját (tehát nem örökölt) attribútumait is; az opcionális attribútumokat a „systemMayContain” tulajdonsághoz kell hozzáadnunk, a kötelező attribútumokat pedig a „systemMustContain” tartalmazza. Az utóbbi tulajdonságot csak az objektum létrehozása közben adhatjuk meg, későbbi módosítására már nincs lehetőség.

Fontos a „subclassOf” tulajdonság megadása, ez határozza meg, hogy új típusunk kitől (illetve kiktől) fog attribútumokat örökölni.

```
DirectoryEntry objNewClass;  
const string strNewClassName =  
% "DotNetFelhasználó";  
string strCNNewClassName = String.Concat("CN=",  
%strNewClassName);  
objNewClass = objSchema.Children.  
%Add(strCNNewClassName, "classSchema");  
//a user osztálytól örökítünk  
objNewClass.Properties["subclassOf"].  
%Add("User");  
//egyedi governsID  
objNewClass.Properties["governID"].  
Add("1.2.840.113556.1.6.1.5.23");  
//leírás  
objNewClass.Properties["adminDescription"].  
%Add("DotNet Séma objektum");  
//a létrehozott attribútum lesz a típus  
%egyik opcionális tulajdonsága  
objNewClass.Properties["systemMayContain"].  
%Add(strNewAttributeName);  
//a módosításokat elküldjük a címártnak  
objNewClass.CommitChanges();  
Console.WriteLine("A létrejött séma objektum:  
%"+objNewClass.Name);  
Console.WriteLine("A "+ objNewClass.Name + "  
%típus, és az "+ objNewAttribute.Name + "  
%attribútum összerendelése megtörtént.");
```

## Objektum létrehozása

Ezután a séma készen áll arra, hogy az alkalmazás partíciókban új „DotNetFelhasználó” típusú objektumokat hozzunk létre. Hogy ezt megteheszünk, szükségünk lesz a címártnak valamelyik alkalmazás partíciójához csatlakozó DirectoryEntry objektumra, ennek „Children” kollekciójához kell hozzáadnunk az új objektumot.

Az alábbi kódrészletben létrehozuk az új felhasználó objektumot, és kitöltjük annak „weblap” tulajdonságát is.

```
string strHostName = GetName("Kérem az ADAM  
%kiszolgáló nevét (alapértelmezés: localhost): ",  
% "localhost");  
string strPortNumber = GetName("Kérem a portszámot
```

```

% (Alapértelmezés: 389): ", "389");
DirectoryEntry objAppPart = new DirectoryEntry
% ("LDAP://" + strHostName + ":" + strPortNumber +
% "/CN=AppPartition, DC=falatrax, DC=hu");
DirectoryEntry objNewUser = objAppPart.Children.
% Add("CN=SziklaSzilard", "DotNetFelhasznalo23");
objNewUser.Properties["weblap"];
% Add("http://weblap.hu");
objNewUser.CommitChanges();
Console.WriteLine(objNewUser.Name);

```

## Új alkalmazás partíció hozzáadása

Természetesen lehetőségünk van arra is, hogy programból hozzunk létre új alkalmazás partíciót a címtárban. Szükségünk lesz egy olyan DirectoryEntry objektumra, amely a címtár gyökerére (vagy más konténer objektumra) mutat; ebben fog létrejönni az alkalmazás partíció. A partíciót reprezentáló DirectoryEntry létrehozásakor a konstruktorban „domainDNS” típust, és az új partíció DN-jét kell megadnunk. A kötelező tulajdonságok megadása után az új objektumot elküldhetjük a címtárnak.

Az alábbi kódrészlet a címtár gyökerében hoz létre új alkalmazás partíciót a már meglévő mellé.

```

static void CreateAppPart() {
    string strHostName = GetName("Kérem az ADAM
% kiszolgáló nevét (alapértelmezés: localhost):
% ", "localhost");
    string strPortNumber = GetName("Kérem a
% portszámot (Alapértelmezés: 389): ", "389");
    string strLDAP = "LDAP://" + strHostName + ":" +
% strPortNumber;
    DirectoryEntry parent = new
% DirectoryEntry(strLDAP);
    Console.WriteLine(strLDAP);
    DirectoryEntry domainDNS = parent.Children.
% Add("CN=AppPartition2, DC=falatrax,
% DC=hu", "domainDNS");
    domainDNS.Properties["instanceType"].Value = 5;
    domainDNS.Properties["objectClass"].Value =
% "container";
    domainDNS.CommitChanges();
}

```

## Objektumok tömeges létrehozása és törlése

Végül próbáljuk ki azt, hogy mennyi ideig tart például 1000 darab új objektum létrehozása a címtárban.

Az alábbi kódrészletben csatlakozunk egy alkalmazás partícióhoz, létrehozunk, majd egyesével feltöltjük az objektumokat, és mérjük az eltelt időt.

```

static void CreateMultipleObjects() {
    DirectoryEntry objNewUser;
    string strHostName = GetName("Kérem az ADAM

```

```

% kiszolgáló nevét (alapértelmezés: localhost): ",
% "localhost");
    string strPortNumber = GetName("Kérem a portszámot
% (Alapértelmezés: 389): ", "389");
    DirectoryEntry objAppPart = new
    DirectoryEntry("LDAP://" + strHostName + ":" +
% strPortNumber + "/CN=AppPartition, DC=falatrax,
% DC=hu");
    DateTime start = DateTime.Now;
    %for (int i=0; i<1000; i++) {
    String userName = "CN=Felhasznalo" + i;
    %objNewUser = objAppPart.Children.Add(userName,
% "DotNetFelhasznalo23");
    Console.WriteLine(userName);
    objNewUser.CommitChanges();
    objNewUser.Close();
    }
    DateTime stop = DateTime.Now;
    TimeSpan duration = stop - start;
    Console.WriteLine(duration);
}

```

Az objektumok törléséhez csak egy helyen kell módosítanunk az előző kódot, a for ciklusban a következő soroknak kell szerepelni:

```

objUser = new DirectoryEntry("LDAP://" + strHostName
% + ":" + strPortNumber + "/" + userName +
% ", CN=AppPartition, DC=falatrax, DC=hu");
objAppPart.Children.Remove(objUser);

```

## Összefoglalás

Amint láthattuk, az ADAM kiválóan alkalmas arra, hogy akár önállóan, de még inkább az Active Directory kiegészítéseként címtárszolgáltatást nyújtson az ezt igénylő alkalmazások számára. Az ADAM címtár sémája akár a vele kapott felügyeleti eszközök segítségével, akár programozott módon könnyen alakítható, így rugalmasan képes alkalmazkodni a legváltozatosabb igényekhez is. A sémamódosítások nem érintik az egész vállalatot (mint az Active Directory esetében), így több lehetőség van a kísérletezésre, egyedi megoldások kipróbálására.

SZERÉNYI LÁSZLÓ  
szerenyi@met.hu

## A cikkben szereplő URL-ek:

[1] [http://store.netacademia.net/mshu/OTHER/  
%technet\\_code/ADAM.zip](http://store.netacademia.net/mshu/OTHER/%technet_code/ADAM.zip)



# RAID-ers

## A DROMEDÁR ESETE A SÍVATAGBAN A RAID-EREKKEL 2. RÉSZ

A kiszolgálókban mindennapos a RAID alkalmazása, gyakran azonban a használói sincsenek egészen tisztában azzal, hogy pontosan mit, illetve miért használnak. Cikkemben összefoglalom mindazt, amit a RAID-ról tudni érdemes. Ebben a részben a Windows szoftveres RAID megoldásait mutatom be. Bár én a magyar nyelvű kifejezéseket használom, a példákat angol nyelvű operációs rendszer segítségével mutatom be, mert a kiszolgálókra leggyakrabban angol nyelvű operációs rendszert telepítenek.

### Soft RAID

A szoftveres RAID nem igényel különleges hardvert. A számítógéphez csatlakoztatjuk a diszketek IDE ATA, SATA, SCSI csatló segítségével. Az operációs rendszerre bízunk a munka dandárját.

Előnye:

- Egyszerű, akár otthoni rendszerek védelme is megoldható ezzel a módszerrel.
- A régebbi, kisebb, elfekvő diszkeiket is hasznosítani tudjuk.

Hátránya:

- Az így készített RAID rendszerek teljesítménye általában elmarad egy hardveres RAID megoldás teljesítményétől.
- A RAID kezeléséhez szükséges erőforrásokat (CPU, memória, I/O) az operációs rendszer biztosítja, elvonva azokat az alkalmazásoktól.
- Vírustámadás, az operációs rendszer hibája a RAID rendszerre is kihatással lehet, így az adatok megsérülhetnek.
- Meghibásodás esetén a helyreállítás nem egyszerű/egyértelmű.

### RAID a Windowsban

Az [1] alatt részletes leírás található a Windows RAID megoldásairól.

Lássuk sorba szedve, hogyan készítsünk Windows alatt szoftveres RAID védelmet.

A Windowsban a szoftveres RAID megoldásokat csak az alábbi operációs rendszerek támogatják:

- Windows 2000/XP Professional
- Windows 2000/2003 szerver (mindegyik verzió) (Azaz MS-DOS, Windows 95, Windows 98, Windows Millennium Edition, Windows NT, vagy Windows XP Home Edition NEM kezeli és nem is olvassa az ilyen diszketeket!!!)

## Hogyan csináljunk RAID-et Windows alatt?

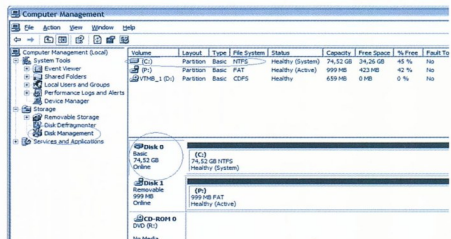
1. Gondoljuk át, milyen RAID megoldást szeretnénk használni.
2. Ennek megfelelő számú diszket installáljunk a gépbe. (min. 2-t.)
3. Ezek után jöhet az operációs rendszer telepítése. (Az operációs rendszer telepítéséhez elegendő egy diszke jelenléte is.) Sőt, meglévő rendszerünket is bővíthetjük újabb diszkekkel, így a meglévő adatainkat is bevitethetjük a RAID-rendszerbe.

4. Ezek után indítsunk el egy lemezkezelőt. (Start->saját gép (jobb gomb)->kezelés. A megjelenő menedzserment konzolon válasszuk a lemezkezelő modult.) (1. ábra)

Biztonsági korlátozások: A legtöbb lemezművelet csak rendszeradminisztrátor jogosultsággal végezhető el.

A telepítés után látható, hogy a diszkek

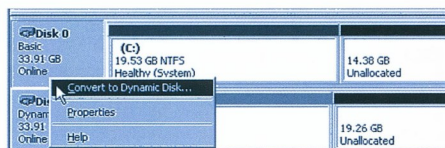
ügynevezett Basic diszkek. Ezekkel egyszerű feladatok végezhetőek el. (Partíciók létrehozása, formázás.)



1. ábra Lemezkezelő

A RAID kötetek elkészítéséhez a lemezt dinamikus diszkké kell konvertálni (2. ábra). Vigyázat, ez egy vissza nem fordítható folyamat, azaz dinamikus diszkeinket (teljes törítés nélkül) nem tudjuk alaplemezzé konvertálni.

Természetesen több diszket kijelölve azokat egyszerre is tudjuk konvertálni (3. ábra). A lemezkezelő egy figyelmeztető ablakban adja tudtunkra, hogy a dinamikus diszkekről (másik partíciójáról) másik operációs rendszer nem tud a jövőben bootolni. A konvertálás során a diszken levő adatok megmaradnak. A konverzió feltétele legalább 1MB üres terület a diszk végén. Ide kerül a dinamikus diszk adatbázisa. Ha Windows-sal partícionáltuk a diszket, akkor a rendszer automatikusan hagyta helyet ennek az adatbázisnak. (Ha más partícionáló programmal készítettük el a partíciót, és nincs hely, akkor advasztás nélkül nem végezhető el a konverzió.) Eltávolítható diszkeket nem lehet dinamikus diszkké konvertálni. Csak 512k szektorméretű diszkek konvertálhatóak dinamikus diszkké. Hordozható számítógépekben nem támogatottak a dinamikus diszkek. (Ha laptopunkban több diszk található, akkor sem használható ez a rendszer dinamikus diszkek kezelésére.)



2. ábra

Néhány további megjegyzés a konverzióval kapcsolatban (ldézet a Windows XP súgóijából)

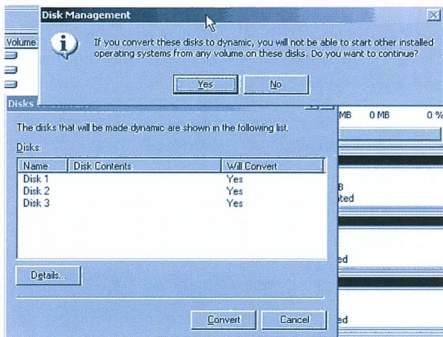
- Alaplemez dinamikus lemezzé történő konvertálása után a dinamikus kötetek nem alakíthatók vissza partíciókká. Ehelyett a lemezen levő valamennyi dinamikus kötetet törölni kell, ezután pedig ki kell adni a `DiskPart convert basic parancsot`. Ha meg szeretné tartani az adatokat, készítsen rólok biztonsági másolatot vagy helyezze át őket másik kötetre. A lemezek konvertálása előtt zárja be a lemezeken futó összes programot.
- A sikeres konvertáláshoz a konvertálni kívánt, fő rendszertöltő rekordot (MBR) tartalmazó lemezekon legalább 1 MB területnek kell lennie a dinamikus lemez adatbázisához. A Windows XP automatikusan fenntartja ezt a területet, amikor egy lemezen partíciókat vagy köteteket hoz létre, de a más operációs rendszerek által létrehozott partíciókon vagy köteteken nem biztos, hogy rendelkezésre áll ez a terület. (Ez a nagyságú szabad terület még akkor is rendelkezésre állhat, ha az a Lemezkezelés beépülő modulban nem látható.)
- A konvertálás után a dinamikus lemezek nem tartalmaznak partíciókat vagy logikai meghajtókat, és a lemezekhez az MS-DOS, Windows 95, Windows 98, Windows Millennium Edition, Windows NT vagy más operációs rendszerek nem férnek hozzá. A dinamikus kötetek elérésére csak a Windows 200X és XP alkalmas.
- Alaplemez dinamikus lemezzé történő konvertálása után az alaplemezen levő partíciók vagy logikai meghajtók a dinamikus lemezen egyszerű kötétté válnak.
- Ne konvertáljon dinamikussá a Windows 200X vagy XP több telepített példányát tartalmazó lemezeket. A lemezek dinamikussá történő konvertálásakor a rendszer áthelyezi a lemez partícióinak minden bejegyzését, kivéve az éppen futó operációs rendszer rendszer-

*és rendszerindító kettőt. A dinamikus lemezzé való konvertálás nem ellenőrizi a Windows többi telepített példányát, és törli a lemezen lévő többi rendszerindító kötet partícióbejegyzéseit. Ezen felül a második példány rendszerleíró adatbázisában a kötéttel kapcsolatos bejegyzések elavulttá válnak, és ennek következtében az operációs rendszer nem indítható el.*

Megjegyzések (Windows XP help):

- **A Windows XP Professional újratelepítése.** A Windows XP Professional csak akkor telepíthető újra egy dinamikus kötetre, ha az adott kötet egy alapkötetből lett konvertálva, és megmaradt a hozzá tartozó bejegyzés a partíciós táblázatban. A Windows XP Professional alatti alaplemezekből konvertált egyszerű kötetekhez nem tartozik bejegyzés a partíciós táblázatban, kivéve, ha a konverzió előtt rendszerkötetek vagy boot kötetek voltak. Ezekre az adatokra szükség van ahhoz, hogy a Windows XP Professional újratelepíthető legyen a dinamikus kötetekre.
- **Sikertelen konvertálás.** Boot lemez konvertálásakor, vagy ha a kötet vagy partíció használatban van a konvertálni kívánt lemezen, a sikeres konvertáláshoz újra kell indítani a számítógépet. A számítógép újraindítása után is sikertelen lehet a konvertálás az alábbi esetekben:
  - Kapcsolat megszakítása minden meglévő dinamikus lemezzel a számítógép újraindítása közben.
  - A konvertálni kívánt lemez vagy lemezek cseréje a számítógép újraindítása közben. (A lemezkezelő érzékelni a lemez cseréjét és a konvertálás megállításulhat.)
  - A konvertálni kívánt lemez kiosztásának módosítása.
  - I/O hiba következik be a lemezen a konvertálási folyamat során.
- **Mikor történik meg a konvertálás?** A rendszer- és boot partíciók a számítógép újraindulása után alakulnak át dinamikus kötetekké. Az összes többi partíció és alapkötet konvertálása azonnali. Ha azonban a konvertálandó lemezen lévő valamelyik partíció vagy alapkötet használatban van, bekövetkezik a „kényszerleállításnak” nevezett esemény, amikor a kötéttel használt programok automatikusan megszakadnak. Ha a kötet esetében a kényszerleállítás nem lehetséges, (például a kötet aktiv paging állományt tartalmaz), akkor a lemez konvertálása csak a számítógép újraindítását követően fejeződik be.
- **Más operációs rendszerek.** Olyan lemez konvertálásakor, amelyen nem Windows 2000 vagy Windows XP Professional operációs rendszer található, a konvertálás után ezzel az operációs rendszerrel nem indítható el a számítógép.
- **Alaplemezek visszaállítás.** Az alaplemez dinamikus lemezzé konvertálása után a dinamikus kötetek nem alakíthatók vissza partíciókká. Ehelyett a lemezzel kell törölni minden dinamikus kötetet, majd a `Convert To Basic Disk` parancsot kell használni.





3. ábra több disk konvertálása és figyelmeztetés

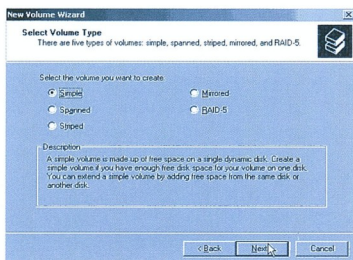
A sikeres konverzió után a diszkeken a partíciók megváltoznak. Helyettük kötetek jönnek létre. A diszken üres helyre kattintva egy varázsló segít nekünk a kívánt kötetek létrehozásában. Lássuk, melyek ezek a kötetek!

## Dinamikus kötetek

A Windows ötféle dinamikus kötetet kezel:

### 1. Egyszerű kötetek (simple volumes)

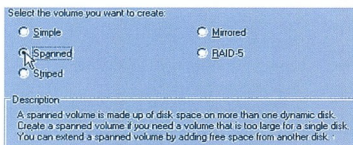
Ez tulajdonképpen az elsődleges partíció dinamikus megfelelője.



4. ábra Egyszerű kötet

### 2. Elosztott kötetek (spanned volumes)

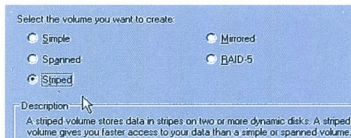
A JBOD megfelelője. Több diszken levő területeket egyetlen logikai meghajtóként látta.



5. ábra Elosztott kötet

### 3. Csíkzott kötetek (striped volumes)

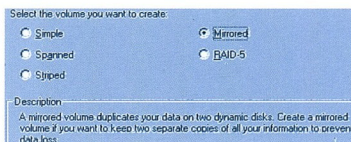
Ez a jó öreg RAID-0. A különböző diszkeken azonos méretű területekből készíthető. Nem hibátűrő megoldás.



6. ábra Csíkzott kötet

### 4. Tükrözött kötetek (mirrored volumes)

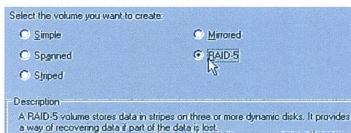
Két különböző diszken azonos méretű kötetekből készül. Ez a RAID-1, hibátűrő megoldás.



7. ábra Tükrözött kötet

### 5. RAID 5 kötetek (RAID 5 volumes)

Ez a hibátűrő megoldás minimum 3 diszkből hozható létre. A diszkeken azonos méretű területeket foglal el.



8. ábra RAID 5

## Csak kiszolgálón használhatunk RAID-1 és RAID-5 köteteket

A tükrözött (RAID-1) és a RAID5 köteteket csak kiszolgáló operációs rendszer (Windows 2000/2003) kezeli. A klienseken (Windows 2000 Professional, XP Home, XP Professional, XP 64bit) csak az első 3 fajta dinamikus kötet (egyszerű, elosztott, csíkzott (RAID-0)) használható, készíthető.

## Parancssor / scriptelés

Természetesen Windows 2k3/XP környezetben minden diszk konfigurációs művelet vezérelhető parancssorból is, a DISKPART program segítségével. Így könnyedén készíthetünk scriptet a diszkek konfigurálására. (Pl.: egy sok gépes környezetben az automatikus telepítési folyamatba beilleszthetőek a diszk konfigurációs műveletek.) Itt látható a diskpart használata parancssorból:

```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.
C:\> diskpart
Microsoft DiskPart version 5.1.3565
Copyright (C) 1999-2003 Microsoft Corporation.
On computer: TEST
DISKPART> list_disk
```

Disk ###	Status	Size	Free	Dyn	Gpt
-----	-----	-----	-----	-----	-----
Disk 0	Online	75 GB		0	B

```
DISKPART> select disk 0
```

Disk 0 is now the selected disk.

```
DISKPART> help
Microsoft DiskPart version 5.1.3565
ADD                - Add a mirror to a simple volume.
ACTIVE             - Marks the current basic partition as an active boot partition.
ASSIGN            - Assign a drive letter or mount point to the selected volume.
BREAK             - Break a mirror set.
CLEAN             - Clear the configuration information, or all information, off the disk.
```

```
CONVERT          - Converts between different disk formats.
```

```
CREATE           - Create a volume or partition.
```

```
DELETE          - Delete an object.
```

```
DETAIL          - Provide details about an object.
```

```
EXIT            - Exit DiskPart
```

```
EXTEND         - Extend a volume.
```

```
HELP           - Prints a list of commands.
```

```
IMPORT         - Imports a disk group.
```

```
LIST           - Prints out a list of objects.
```

```
INACTIVE       - Marks the current basic partition as an inactive partition.
```

```
ONLINE         - Online a disk that is currently marked as offline.
```

```
REM            - Does nothing. Used to comment scripts.
```

```
REMOVE         - Remove a drive letter or mount point assignment.
```

```
REPAIR         - Repair a RAID-5 volume.
```

```
RESCAN        - Rescan the computer looking for disks and volumes.
```

```
RETAIN         - Place a retainer partition under a simple volume.
```

```
SELECT        - Move the focus to an object.
```

```
DISKPART> detail
```

Microsoft DiskPart version 5.1.3565

```
DISK          - Print a list of volumes on the disk.
PARTITION     - Shows details about the current partition.
VOLUME        - Print a list of disks that contribute to the volume.
```

```
DISKPART> detail disk
```

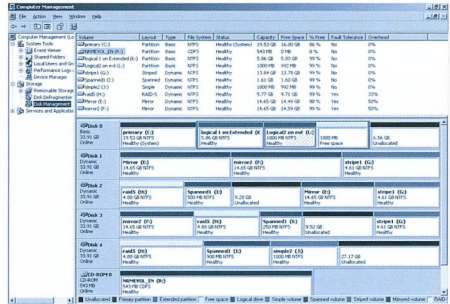
```
HTS548080M9AT00
Disk ID: A8D32665
Type : IDE
Bus : 0
Target : 0
LUN ID : 0
```

Volume ###	Ltr	Label	Fs	Type
Volume 2	C		NTFS	Partition
75 GB	Healthy	System		

## Az állatorvosi dromedár

Érdekességképpen készítettém egy konfigurációt (9. ábra). Ennek a szegény állatnak minden pupja megvan. Mindenféle és fajta alap és dinamikus diszk konfiguráció szerepel rajta. Néhány érdekes dolog szembőllik a képet nézgetve:

- Egy diszken teljesen függetlenül többféle kötetet is létrehozhatunk.
- A grafikus megjelenéskor nem méretarányosan ábrázolja a diszkeken a különböző köteteket.
- Nem látható egyértelműen, hogy melyek az összetartozó kötetek. (Az elnevezések azért sokat segíthetnek, ezért használjunk beszédes neveket!)



9. ábra Állatorvosi dromedár diszk konfiguráció

Nézzük most meg részletesebben az egyes kötetek kezelését, használatát.

## Alap [basic] diszk:

Csak elsődleges vagy kiterjesztett (extended) partíciókat hozhatunk létre. A kiterjesztett partíción több logikai diszket is definiálhatunk.

### 1. Egyszerű kötet

Ez a legegyszerűbb kötet, ami létrehozható dinamikus diszken. Egybefüggő lemezterület. Lehet FAT vagy NTFS formátumban formázva. Meghajtó betűjel adható neki. Önmagában nem sokkal használhatóbbra, mint az elsődleges partíció. A fő felhasználási területe a jövőjében rejlik. Továbbfejleszhető másik kötétté, adatvesztés nélkül.

#### ■ Bővítés

Ekkor a kötet a diszken bővíthető adatvesztés nélkül, a meglévő üres helyet felhasználva. Csak NTFS kötet bővíthető. Rendszer és boot kötet nem bővíthető.

#### ■ Elosztott kötétté bővítés

Ha a kötéthez másik diszken levő üres helyeket is hozzáadunk, akkor elosztott kötétté válik.

#### ■ Tükör készítés

Ha van egy másik diszken elegendő üres helyünk, akkor tükörözhetjük azt a kötetet, ekkor a típusa tükörözött kötétté változik. (11. ábra)

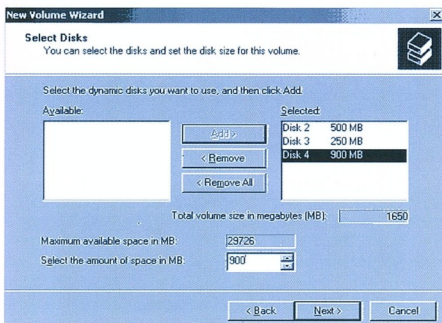


## 2. Elosztott kötet

A JBOD Windows megfelelője.

Segítségével több kisebb diszket használhatunk egyetlen logikai diszkként. A különféle diszkeken különböző méretű területeket használhatunk fel. Természetesen nem hibátűrő megoldás. Így igényeinknek megfelelően hatalmas méretű, összefüggő diszke területeket hozhatunk létre különböző méretű és fájtájú diszkek felhasználásával. A RAID-0-tól eltérően az operációs rendszer folyamatosan tölti meg kötetet, azaz szépen sorbaállítja a diszkeket és egyszerre csak egy diszke-re ír, amíg azon meg nem telik a terület, utána folytatja a következő diszken az írást. Lehet FAT vagy NTFS formátumban formázva. Meghajtó betűjel adható neki.

- Bővítés  
Csak NTFS kötet bővíthető.  
Rendszer és boot kötet nem bővíthető.
- Nem tükrözhető.
- Nem lehet RAID-0-ba illeszteni.
- Nem lehet RAID-5-nek a tagja.



10. ábra Elosztott kötet készítése

## 3. Csikózás

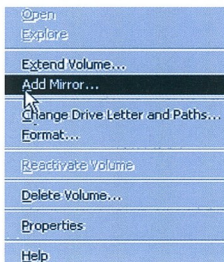
Azonos méretű területekből különböző fizikai diszkeken hozható létre. Az elosztott kötethez képest itt az adatokat csikóban helyezel el az operációs rendszer a diszkeken, így egy adatsomag írása/olvasása egyszerre több diszke történik. Ezzel a megoldással lényegesen gyorsabb virtuális diszkek készíthetőek, mint az építőegységekként felhasznált fizikai diszkek. Minimálisan két diszkből készíthető, több diszket használva a teljesítmény növelhető. Ha a diszkeket külön vezérlőkre kötjük, akkor a teljesítmény tovább növelhető. Nem hibátűrő megoldások, azaz a tömb bármelyik tagjának meghibásodása esetén a teljes logikai diszke tartalma elvész.

- A csikozott kötetek nem tükrözhetőek.
- A kötetek nem bővíthetőek

## 4. Tükrözés

A tükrözés a RAID-1-et valószínűleg meg. Az adatok 2 példányban kerülnek kiírásra, így az egyik diszke meghibásodása esetén az adatok a másik diszken is rendelkezésre állnak. A rendelkezésre állás növelhető, ha a tükörpár tagjait külön diszke-vezérlőkkel hajtjuk meg, ekkor a vezérlő meghibásodásakor is elérhetőek maradnak az adatok a másik diszken. Ezt a konfigurációt duplex diszke konfigurációnak nevezzük. A dupla írási művelet lassítja a rendszer teljesítményét. A Windows aktív

tükört készíti, azaz olvasáskor a tükörpár mindkét tagjáról képes olvasni adatokat, így gyorsítja a logikai lemez adatainak elérési sebességét. Mind az írási, mind az olvasási műveletek gyorsíthatóak duplex diszke konfiguráció használatával. Természetesen boot- és rendszerpartíciók is tükrözhetőek.



11. ábra Tükör hozzáadása

Egyszerűen használható szoftveres megoldás, viszonylag kevés rendszererőforrást használ. Meghibásodás esetén nem romlik lényegesen a rendszer teljesítménye. (Az adatok elérhetőek a másik diszkről.)

## Mi a teendő, ha megsérül a tükör?

Még mielőtt éles adatokkal töltenénk fel, célszerű kipróbálni, hogy az egyik diszket kivéve a rendszer elindul-e a másik diszke-vel. (Ne feledjük el megismételni a tesztet a másik diszket kivéve is!) Erre elsősorban a hardveres konfigurációk különbözősége miatt lehet szükség. Ki kell tapasztalni, hogyan lehet kiiktatni a rendszerből a meghibásodott diszket, illetve hogyan lehet kicserélni a hibás eszközt. (pl.: master/slave, vagy drive ID beállításra, átállításra lehet szükség, hogy megfelelő legyen a hardver konfiguráció, bootolni tudjon az operációs rendszer.) Bootolás után a lemezkezelőt indítsuk el. Össze kell törnünk a tükört. Ezzel a művelettel a tükrözött területből egyszerű kötetet készíthetünk.

A megfelelő tükrözött partícióra kattintva választhatjuk a tükör összetörése (break mirror) parancsot. A művelet befejezése után egyszerű kötet lesz a tükrözött partícióból. Az új, behelyezett diszket dinamikus diszke-ke konvertáljuk. Ezek után úgy járunk el, mint kezdetben a tükör létrehozásakor, egy másik diszken lévő üres területre egy új tükört készíthetünk.

## 5. RAID-5

RAID-5 hibátűrő diszke kötet minimálisan 3 fizikai diszkből hozható létre. Egyforma méretű területekből készíthető. Jobb a helykihasználása, mint a tükrözött kötetnek. A kötetből egy diszke helyet foglal el a paritásadat. (3 darab 100 Gb-s diszke-területből 200 Gb logikai diszke készíthető.)

- A RAID-5 kötetek nem bővíthetőek és nem tükrözhetőek.

A használatához memóriára, illetve íráskor a paritásadatok számítására CPU-teljesítményre van szükség. Meghibásodás esetén a teljesítménye jelentősen romlik. Minden egyes olvasáskor a hiányzó adatokat a paritásból újra számolni kell, ez memóriát és CPU időt igényel.

## Hibaelhárítás

Derítsük ki, mi okozza a hibát. (A részletes hibakereséssel ebben a cikkben terjedelmi okokból nem foglalkozom.) Cseréljük ki a meghibásodott hardverelemet (diszket / vezérlőt, stb.) Miután kicseréltük a meghibásodott diszket, bootoljunk, majd indítsuk el a lemezkezelőt. A RAID-5 egyik kötetén megjelenik egy új menüpont: Repair Volume, ezt választva, majd kijelölve az üres területet a regeneráció elkezdődik. (A menüpont csak akkor jelenik meg, ha van egy olyan üres lemezterület egy dinamikus diszken, amely legalább akkora, mint amekkora a RAID-5 kötet mérete.) A regenerálás is CPU- és memóriaigényes művelet. Lehetőleg olyan időszak-

ban végezzük, amikor a rendszer nem nagyon terhelte. A sikeres regeneráció végén a RAID 5 kötet státusza újra Healthy lesz.

MEGYESI BARNABÁS  
megyesi.barnabas@flag.hu  
MCSE, MCT, HP ASE, dromedárszakértő

### A cikkben szereplő URL-ek:

- [1] <http://www.microsoft.com/technet/prodtechnol/win-dowsserver2003/library/ServerHelp/cc6970e3-d877-4514-b5bf-67b6a80f30f2.mspx>

# Microsoft CRM 3.0

A Microsoft CRM új verziója kibővített megoldáskészletet biztosít a már megszokott Microsoft Office és Microsoft Outlook környezetben.

**A** Microsoft CRM 3.0 marketing, értékesítési és ügyfélszolgálati funkciókat biztosít, a Microsoft Office és Microsoft Outlook programoknál már megszokott felhasználói környezetben. Az új, széleskörű beállítási, testreszabási és integrációs lehetőségek megkönnyítik az ügyfelek és partnerek számára a személyre szabott megoldások bevezetését.

Az előre elkészített jelentések és nézetek használatával az ügyfelek azonnal láthatják az eredményeket és a problémákat, a valós idejű CRM adatokat pedig közvetlenül a Microsoft Excelben elemezhetik, függetlenül attól, hogy kapcsolódnak-e a hálózathoz vagy sem.

A Microsoft CRM 3.0 tökéletesen kiegészíti a CRM eszközöket a listák, kampányok, marketing erőforrások és zárt rendszerű visszacsatolások kezelésére szolgáló marketing-automatizáló modul révén. Az új verzió emellett egy kifinomult szolgáltatás-ütemező modult is tartalmaz, amely automatikusan kezeli azokat az összetett ütemezési kéréseket, amelyekhez jelenleg emberekre, meghatározott készségekre és erőforrásokra van szükség. Az új modulok, illetve már a Microsoft CRM 1.2 verzióban is megtalálható, és továbbfejlesztett értékesítési és ügyfélszolgálati lehetőségek nagymértékben konfigurálhatók és a munkafolyamatok által vezérelhetők, ami biztosítja a CRM üzleti folyamatok konzisztens végrehajtását, a kivételek, hibák, illetve riasztások automatizált kezelését, és az egész szervezetre kiterjedő, zárt rendszerű jelentéskészítést illetve elemzést tesz lehetővé.

Az új verzió tovább bővíti a Microsoft CRM szolgáltatásközpontú architektúrájának konfigurálási, testreszabási és integrációs lehetőségeit. A rendszerhez könnyedén hozzáadhatók új adat-objektumok, melyek előre megadott objektumokhoz csatolhatók, miközben a rendszer a háttérben automatikusan gondoskodik minden adattárolási és webes szolgáltatásról. Az új Környezetdiagnosztikai Varázsló (Environmental Diagnostics Wizard) több mint 100 rendszer- és hálózati beállítást ellenőriz automatikusan, hogy gyors és megbízható telepítést biztosítson, az új Frissítési Tanácsadó (Upgrade Advisor) pedig leegyszerűsíti a frissítési folyamatot a Microsoft CRM 1.0 és 1.2 ügyfelek számára. A Microsoft Operations Manager alkalmazással való integrálás a teljes rendszer felügyeletét megkönnyíti.

A Microsoft CRM 2002-es megjelenése óta világszerte több mint 4000 vállalatnak segít üzleti mutatókkal is mérhető fejlődést elérni ügyfeleikkel kapcsolatos folyamataikban. A magyar nyelvű változat, 2004. szeptemberi megjelenése óta közel 20 vállalat választotta ügyfélkapcsolat-kezelési rendszerül a Microsoft CRM-et. Ezek között vannak 70-nél is több felhasználót kiszolgáló rendszerek, de olyanok is, amelyek csak csupán 1-2 munkatárs használ. Referenciáink között kereskedő, szolgáltató, tanácsadó, oktató cégek is megtalálhatók.

A Microsoft CRM 3.0, a Microsoft CRM jelenlegi felhasználói számára 2005. negyedik negyedévében lesz elérhető, magyar nyelven pedig majd 2006. első negyedévében jelenik meg.



# Windows szolgáltatók 8.rész

## TOVÁBBRA IS A LOCALSYSTEM FIÓK HASZNÁLATÁRÓL

Már csak két kupac szolgáltatás maradt ismertetésre, ám ezek a lista végén szereplők sem kevésbé fontosak/érdekesek, mint az eddigiek.

### Special Administration Console Helper (Speciális felügyeleti konzol segédalkalmazása)

A szerviz rövid neve: Sacsvr

Az alkalmazás neve: sacsvr.dll (svchost.exe)

Függés: -

Függesztés: -

Porthasználat: -

Alapértelmezett indítás: kézi, leállítva

Ez a szerviz egy nagyon érdekes területen fejt ki hatását, az Emergency Management Services („Válságkezelési szolgáltatások”) eszközcsoporthoz tartozik. Ez a haladó kategória a távmenedzsmenst illetve a súlyosabb rendszerproblémák elhárításával kapcsolatban (az alap kategóriába én a boot menü eszközeit és a Recovery Console-t sorolom). Ha nincs hálózat, nincs internet és semmilyen más, hagyományos módon (telnet, RD) nem lehet elérni az adott szervert, csak pl. a soros porton egy nullkábellel, akkor jöhetnek ezek az eszközök. Jó tudni, hogy a Windows operációs rendszerek következő részei működnek együtt az EMS-sel:

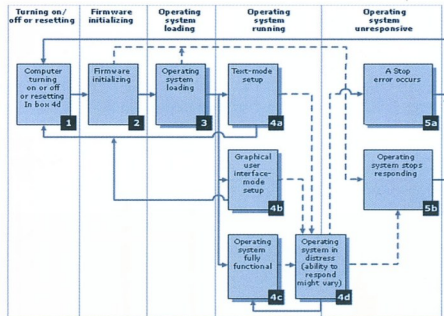
- a telepítő betöltőprogramja és a text módú telepítő
- az Ntldr
- az Ntoskrnl.exe
- a Cmd.exe
- a Recovery Console
- a RIS (azaz csak egy spéci Startrom.com változat)
- és a STOP hibákkal kapcsolatos üzenetek

Az együttműködés konkrétan annyit jelent, hogy például ha már az Ntldr fut, akkor a távoli gépről, a soros porton keresztül „távirányítással” választhatunk a gépre telepített operációs rendszerek közül, illetve akár indíthatjuk a Recovery Console-t is. Ez úgy lehetséges, hogy a problémás gép videokimenete sima text formátumban megjelenik a soros porton, és ezután a másik gép monitorán. A távoli billentyűzetről bevitt adatokat ekkor visszafelé – szintén a soros porton keresztül - konvertálva tudja fogadni a gép, így innen-től a konzol az úr, ergo megtörtént a konzolátírányítás.

Az EMS három részből áll, a már említett konzolátírányításból és az erre épülő két további komponensből, a Speciális felügyeleti konzolból (SAC azaz) illetve a !SAC-ból, ami pedig a SAC alternatívája.

A SAC-ot elsőre egy parancssorhoz lehetne hasonlítani, ám valójában nincs közé hozzá, és a lehetőségei sem érnek azért fel vele. Amit biztosan tud:

- gép leállítása és újraindítása
- a processzek/szervizek listázása,
- a processzek lezárása, prioritás változtatása
- a szerver IP címének ellenőrzése és beállítása
- dátum és idő beállítása
- parancssor elindítása



■ Az árnyékolt téglalapok jelzik a SAC lehetséges működési területeit

A !SAC egy kritikusabb helyzetre tervezett eszköz, független a SAC-tól, ráadásul a váltás a két komponens között szituációtól függően automatikus, illetve ha a SAC betöltése semmiképp nem megy, akkor is a !SAC kerül meghívásra. Két fontos feladata van, a STOP üzenetek átírányítása, illetve az újraindítás a SAC konzol kiesése esetén.

Mivel a szerviz a SAC szolgáltatások kiszolgálója, ha letiltjuk, ezek a komponensek nem lesznek képesek működni, de mivel ritkán (vagy az is lehet hogy soha) lesz rá szükség maradjon tehát alapértelmezés szerint leállítva.

### System Event Notification (Rendszeresemény jelzése)

A szerviz rövid neve: Sens

Az alkalmazás neve: sens.dll (svchost.exe)

Függés: COM+ Event System, Remote Procedure Call

Függesztés: -

Porthasználat: -

Alapértelmezett indítás: automatikus

A Windows rendszeresemények jelzőberendezése, pl. a hálózattal vagy az energiaellátással kapcsolatos események jelzésének közvetítője, de pl. az Exchange Server is használja ugyanerre a célra. A szolgáltatás értesíti a COM+ Event System szervizt, ami aztán továbbadja az erre „feliratkozók” (subscribers) számára az üzeneteket.

Ha letiltjuk, akkor a következő konkrét jelenségekkel biztosan kell számolnunk:

- a SENS interfész leáll, a SENS Logon/Logoff üzeneteinek vége szakad
- két, a mobil alkalmazásokkal illetve a laptopokkal kapcsolatos speciális Win32 API (IsNetworkAlive()) illetve a IsDestinationReachable()) sem működik tovább
- a SyncMgr (Mobsync.exe) szinten befejezi pályafutását, azaz a különböző hálózattal, hálózati kapcsolódással kapcsolatos üzenetek továbbítása is véget ér.

### Task Scheduler (Feladatütemező)

A szerviz rövid neve: Schedule

Az alkalmazás neve: schedsvc.dll (svchost.exe)

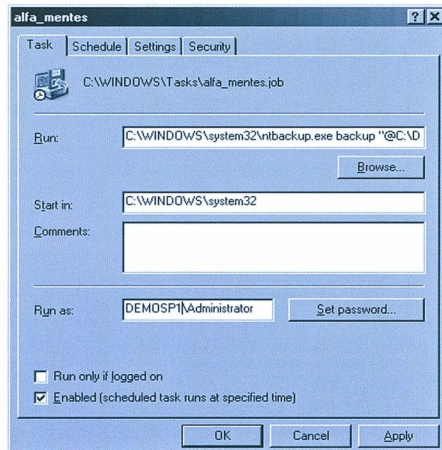
Függés: Remote Procedure Call

Függesztés: -

Porthasználat: -

Alapértelmezett indítás: automatikus

A Feladatütemező szintén hasznos eszköz, korrekt kiegészítő például az NTBackup-nak, de sok más esetben is jól használható. Funkciója egyértelmű: bármilyen szkript, alkalmazás vagy akár dokumentum időzített futtatását oldhatjuk meg vele. Az általunk kreált feladatokat (.job) a Windows\Tasks mappába menti. Az adott feladat beállításai az elkészítésekör rögzülnek, ergo bármilyen globális beállítás ezekre kevés hatással lesz.



### Egy szimpla feladat jellemzői

Visszont egy-egy feladat másolása gépek között triviális, a job állományt kell csak a fent említett mappába másolnunk és máris kész. Pontosabban a jogosultságot (felhasználónevet/jelszót) újra be kell állítanunk ekkor ugyanúgy, mint bármely más alkalommal, amikor a már létező feladat tulajdonságaiba belepiszkálunk.

Ha már a jogosultságoknál tartunk, jó tudni, hogy a Windows Server 2003 telepítése után ez a szerviz automatikusan elindul, ám alapértelmezés szerint csak az Administrators, a Backup Operators, vagy a Server Operators csoport tagjaként leszünk képesek időzített feladatokat gyártani/kijavítani/törölni/leállítani és elindítani. Ettől függetlenül egy Backup Operator nem adhat meg egy rendszergazda fiókot a feladat „futtatójaként”, fordítva viszont természetesen igen. Ha mégis szükség van a fenti hármas hozzáférés kibővítésére más csoportoknak is, akkor belefutunk egy érdekes szituációba: nincs Security/Sharing stb. fül a mappa tulajdonságai között, csak egy árva General árvalkodik a szokásos 4-5 helyett (persze ha van IIS és azonos partíción van, akkor egy Web Sharing fül azért lesz rajta ©). Akkor hogyan tovább? Nincs veszve semmi, jöhet a parancssor, és például a „cacls” parancs, amellyel hozzáadhatjuk a jogosultsági körhöz a kívánt csoportot/felhasználót (ellenben ilyenkor a „többiek” által kreált feladatokhoz is hozzá fog férni, szóval csak óvatosan ezzel a lehetőséggel).



### ■ Kissé hiányos, de nem véletlenül...

Amint még az időzített feladatokkal kapcsolatban ne felejtünk el (sűrűn visszaüt): a jogosultsághoz megadott jelszó nem szinkronizálódik, azaz ha az adott feladatot futtató felhasználó jelszava változik, manuálisan utána kell állítanunk. Amennyiben a szerviz leállítjuk, az időzített feladatok értelemszerűen nem állnak tovább rendelkezésre. Ám ha nincs rá szükségünk, bátran leállíthatjuk (de ne feledjük: pl. a mentésnél is szükség lehet rá).



## Telephony (Tárcsázó)

A szerviz rövid neve: TapiSrv

Az alkalmazás neve: tapisrv.dll (svchost.exe)

Függés: Remote Procedure Call, Plug and Play

Függésztés: Remote Access Connection Manager, Remote Access Auto Connection Manager, Internet Connection Firewall (ICF) / Internet Connection Sharing (ICS), Fax

Porthasználat: -

Alapértelmezett indítás: kézi, leállítva

Ez a szerviz a „Tárcsázó” alkalmazás működését biztosítja, ami egy TAPI (Telephony API) alapú program. Ennek segítségével telefonálhatunk a számítógéphez csatlakoztatott modem keresztül, illetve IP alapú telefonhívásokat is kezdeményezhetünk hálózatunkon belül. Valamint további speciális (PSTN illetve internetes IP alapú) telefonhívásokat is bonyolítunk pl. egy a telefonközpontunkra kötött H.323 proxy/átjáró gép segítségével, ráadásul IP alapú videokonferenciák megteremtését is megvalósíthatja. A programot kiszolgáló szervizt nem lehet leállítani abban az esetben, ha egy tőle függő szerviz elindult. Ha nincs aktív állapotú függő szerviz, és a Tárcsázó szolgáltatás le van állítva, akkor a függő szervizek a különböző alkalmazások (pl. a Windows Product Activation ©) kérésére képekkel elindítani. Ha letiltjuk, akkor a függő szolgáltatásokon és a kapcsolódó alkalmazásokon túl pl. az egész modernes alrendszer is megbénul. Mérlepeljünk!

## Terminal Services

(Terminálszolgáltatások)

A szerviz rövid neve: TermService

Az alkalmazás neve: termsrv.dll (svchost.exe)

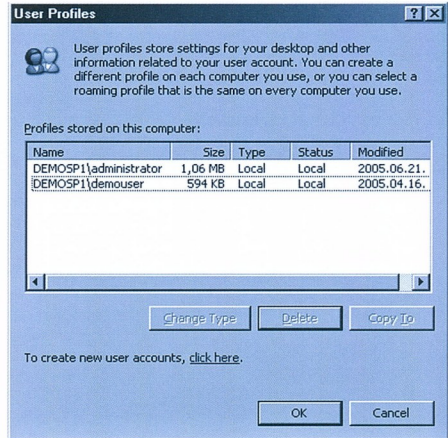
Függés: Remote Procedure Call, Infrared Monitor

Függésztés: Fast User Switching Compatibility Services

Porthasználat: TCP: 3389, UDP: 1036, RDP: 3389

Alapértelmezett indítás: kézi, leállítva

Jól ismerjük: már az NT4 Terminal Server Edition óta jóbarát és hasznos eszköz rengeteg szituációban. A Windows 2003-ban a Terminal Service alapértelmezés szerint remote administrator módban üzemel, ergo két, szimultán kapcsolat minden beállítás és licenc nélkül megengedett az Administrators csoport tagjai számára (vagy akit behelyezünk pluszban a Remote Desktop Users csoportba). Ha ennél többet akarunk, akkor a szokásos módon kell telepítenünk a „rendes” terminál szervert. Mivel a TS nagy falat és ebben a sorozatban csak a szervizről szólnunk, egyetlen gyakorlati tapasztalatot alapuló észrevételt szeretnék megemlíteni. A Windows 2000-el ellentétben a szerviz akadályai helyett az előző ábrán látható helyről induló szokványos profilmasolási procedúrának. Tudniillik az XP/2003 esetében szoros kapcsolatban áll a „Fast User Switching” szolgáltatással, ergo nem engedni a profil ráhúzást a rendszer egy rezidens felhasználójára. Erről (persze nem egyértelműen ezt jelezve) egy kedves hibaüzenet formájában tudósít is. A megoldás a TS szerviz leállítása lehetne, amelyet viszont nem tudunk elvégezni a Services MMC-ből, így csak a leállítás és a gép újraindítása marad, mint szükséges előjáték a profilmasolás előtt.



## Terminal Services Session Directory

(Terminálszolgáltatások munkamenetköltárné)

A szerviz rövid neve: Tssdls

Az alkalmazás neve: tssdls.exe

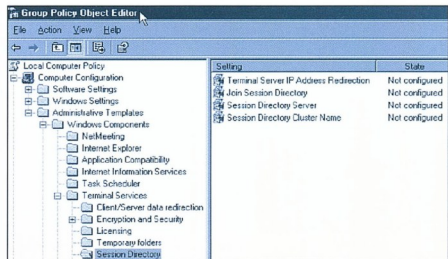
Függés: Remote Procedure Call

Függésztés: -

Porthasználat: UDP: 1083

Alapértelmezett indítás: leltitva

Csak speciális esetben szükséges, mégpedig akkor ha a terminál szervereinke a stabilitás és a nagyobb teljesítmény elérése érdekében fűrtbe szeretnénk kötni.



## A helyi házirendben is megtalálhatóak a TSSD beállításai

Ebben az esetben a felhasználók kapcsolódása a szokásos TCP/IP alapú terheléelosztás módszerrel történhet meg, azaz mindig az első, az adott pillanatban „ráérró” kiszolgáló kapja a kérést. Ha eddig egy máson futott a felhasználó munkamenete (és mondjuk megszakadt), akkor a munkamenetköltárné bejegyzése alapján az eddigi kiszolgálóhoz kerül vissza. Ehhez szükséges ez a szerviz, amelyet természetesen a fűrt összes gépén el kell indítanunk. Ezután a Csoport/Helyi házirendben (vagy a TS admin eszközében) be kell állítanunk ennek a speciális adatbázisnak a jellemzőit az

előző ábrán látható helyen. (A [2] címen rengeteg ide passzoló szkriptet találunk.)

Ha ezt a szervizt leállítjuk vagy letiltjuk, akkor a terhelésselosztás nem fog működni, más ismert hatása viszont nincs.

## Uninterruptible Power Supply

(Szünetmentes áramforrás)

A szerviz rövid neve: UPS

Az alkalmazás neve: ups.exe

Függés: -

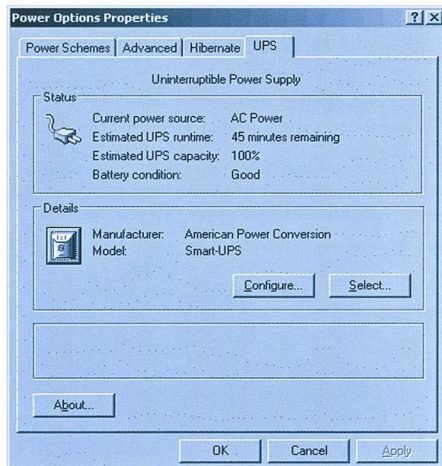
Függesztés: -

Porthasználat: -

Alapértelmezett indítás: kézi, leállítva (lásd lentebb)

A soros porton keresztül kommunikációt biztosítja a menedzselhető UPS-ekkel. Amennyiben USB-s szünetmentesünk van, akkor viszont bátran leltíthatjuk a szervizt, mert ekkor hatastalan. Alapértelmezés szerint kézi indítású, ám automatikusra vált, ha érzékeli, hogy egy szünetmentest csatlakoztatunk a soros porton.

Ha egy - a listájában szereplő - ismert UPS-ről van szó, akkor a szokásos esemény triggerek kivételével a maximális szinten képes kommunikálni, azaz pl. futási időt, kapacitást, akkumulátor állapotot is jelez, külön vezérlő szoftver nélkül.



### ☞ Nem kell külön vezérlőszoftver

Ha leállítjuk vagy letiltjuk, akkor a kommunikáció az UPS-sel megszűnik, más egyéb ismert hatása nincs.

## Themes (Témák)

A szerviz rövid neve: Themes

Az alkalmazás neve: termsrv.dll (svchost.exe)

Függés: -

Függesztés: -

Porthasználat: -

Alapértelmezett indítás: letiltva

Hogysimondjam? Egy kevésbé jelentős, ám a felhasználói élmény fokozását elősegítő szolgáltatásról van szó. Egy-egy téma a GUI-n előforduló ikonok, betűtípusok, színsémák és hangok csoportját jelenti és ennek a szerviznek a segítségével cserélgethetjük vagy hozhatjuk létre a sajátunkat.

Technikai érdekesség még, hogy a szerviz a Windows 2003-on alapesetben le van tiltva, ekkor nem is válthatunk át a „Windows XP” stílusra. Amennyiben engedélyezzük, akkor igen, ám ha ezután újra leállítjuk vagy letiltjuk, akkor automatikusan átvált a Windows Classic témára.

## Upload Manager (Feltöltéskezelő)

A szerviz rövid neve: Uploadmgr

Az alkalmazás neve: pchsvc.dll (svchost.exe)

Függés: Remote Procedure Call

Függesztés: -

Porthasználat: TCP: 80

Alapértelmezett indítás: kézi, leállítva a Web, Standard, Enterprise verziókon, letiltva a Datacenter Server-en

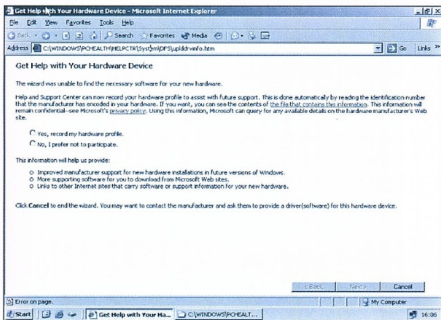
Az Upload Manager szolgáltatás az újfajta hibakeresést illetve támogatást is megvalósító Sűgő rész. Biztosítja a szinkron és aszinkron hálózati fájlátvitelt a Microsoft és az adott gép között. Erre akkor van például szükség, ha az ún. Microsoft Driver Feedback Server-rel kerülünk kapcsolatba, azaz a rendszerbe telepített hardver eszközök meghajtóiról – név nélkül – adunk információt a Microsoftnak, azért hogy az új hardverhez megfelelő illesztőprogramot szerezzünk. Ehhez az operációs rendszer az Upldrinfo.htm állományt használja, amelyet itt találunk:

C:\WINDOWS\PCHEALTH\HELPCTR\System\DFS\

A viszony kölcsönös, hiszen közvetve a MDFE megvizsgálja és le is menti a hardverprofilt, majd ezek után a hardver eszközök frissebb/megfelelőbb verzióit ajánlja letöltésre akár közvetlenül a Microsofttól, vagy a „harmadik” gyártóktól. Nem kell pánikba esni, a rendelkezésre álló információk szerint csak az adott hardver egyedi azonosítóját, a Hardvervárázsló befejezésének időpontját, valamint az operációs rendszer egyedi azonosítóját kapja meg a Microsoft, a felhasználói, gépnév, IP vagy más adatokat nem. Ha viszont letiltjuk, akkor ez a fajta információ küldés/fogadás leáll, más ismert hatása nincs.

(Update: A helyzet az, hogy egyébként sem kell aggódnunk, mivel a fentiek a Windows 2003 SP1 telepítése után érvénytelenné válnak. A Microsoft ugyanis nem használja többé ezt a funkciót. A frissítés után a szerviz letiltásra kerül, ha pedig tiszta SP1-es telepítést hajtunk végre, akkor már nem is kerül be a szerverek közé.)





**Így néz ki az említett oldal**

## Volume Shadow Copy

[Kétpillanatkép]

A szerviz rövid neve: VSS

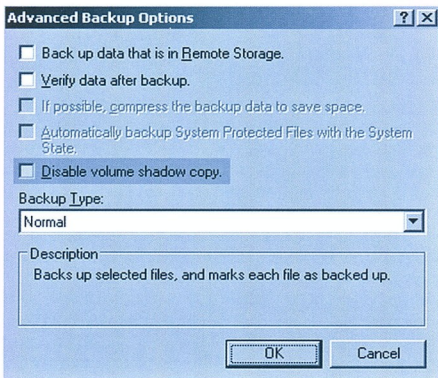
Az alkalmazás neve: vssvc.exe

Függés: Remote Procedure Call

Fűglesztés: -

Porthasználat: -

Alapértelmezett indítás: kézi, leállítva



**Ha itt letiltjuk, előfordulhat, hogy a mentésből kimaradnak az éppen nyitva lévő vagy használt fájlok**

A Windows Server 2003-ban az árnymásolattal kapcsolatos lehetőségek több részből állnak. Használhatjuk a megosztott mappákban elhelyezett állományokon a korábbi verziók vagy a törölt példányok visszaállítására, vagy betöltheti például a mentést és visszaállítást végző alkalmazások kiegészítőjének szerepét. E szerviz az utóbbi funkcióhoz köthetik, azaz ha pl. az NTBackup-nak a visszaállításnál az állomány egy régebbi verziójára van szüksége, akkor ezen szervizen keresztül keresi és kapja meg azt. Abban az esetben szintén segít, ha egy mentendő állomány éppen használatban van, ugyanis ezzel a tudással felvértelve az NTBackup képes arra, hogy korrekelt (árnymá)solatot

(illetve akár többet is egy-egy állományról) készítsen ezekről az állományokról is, miközben egyúttal használatban is vannak.

Ha a szervizt leállítjuk vagy letiltjuk, akkor a mentőszoftverek nem képesek használni ezt a kiegészítő funkciót.

## Virtual Disk Service

[Virtuális lemez szolgáltatás]

A szerviz rövid neve: VDS

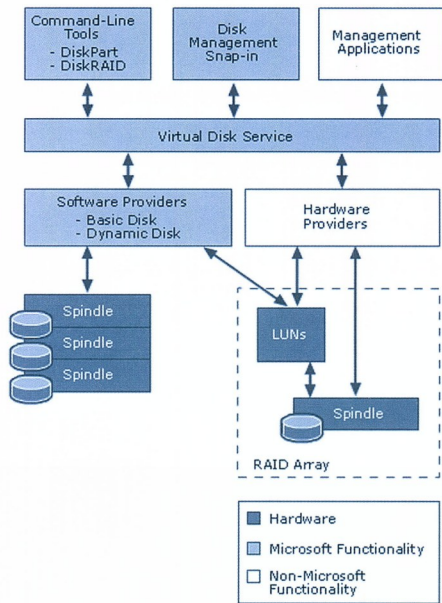
Az alkalmazás neve: VDS.EXE

Függés: Remote Procedure Call, Plug and Play

Fűglesztés: -

Porthasználat: -

Alapértelmezett indítás: kézi, leállítva



**A VDS architektúra**

A Windows 2000-nél a különböző háttértárolással kapcsolatos különleges eszközök (spéci HDD-k, PCI RAID kártyák, lemeztömbök, stb.) esetén minden eszköznek csak a saját gyártótól származó meghajtója és kezelőprogramja volt elérhető számunkra, az operációs rendszer nem egy közös helyen „tartotta kézbén” ezeket. A közös helyen történő felügyelet érdekében a Windows 2003-ba beépítettek egy koordináló eszközt, ez a VDS. Gyakorlatilag egy „sereg API-t” jelent egyetlen kezelőfelületbe építve, ami megoldja a szétszórtság problémáját. A VDS nem egy újabb alkalmazás, hanem egy illesztőfelület, így azok a Windows 2003 komponensek, amelyek hozzá szeretnének férni, két-három további gyárilag beépített illetve az egyéb külső alkalmazásokon keresztül tehetnek meg ezt.

Az egyik a jól ismert Disk Management MMC bővítmény, egy másik a DiskPart.exe, illetve ide tartozik például a Diskraid.exe, ami egy parancsori RAID segédprogram Windows Server 2003 Resource Kit Tools-ból [1].

Ha a szervizt letiltjuk, akkor a VDS szolgáltatás leáll, amelynek (ha használatban van) súlyos következményei is lehetnek.

### Windows Audio (Windows audió)

A szerviz rövid neve: *Audiosrv*  
Az alkalmazás neve: *audiosrv.dll (svchost.exe)*  
Függés: *Plug and Play, Remote Procedure Call*  
Függésztés: -  
Porthasználat: -

Alapértelmezett indítás: *automatikus a Standard verziónál, illetve leiltott a Web, Enterprise és Datacenter változatoknál.*

Szintén a felhasználói élményfokozó szolgáltatások közé tartozik, értelemszerűen a Windows alatti hanggal és zenével kapcsolatos eszközök kezelése tartozik hozzá, pl. figyelni a Plug and Play eseményeket (hangkártyák) vagy mendedszeli az audio API-k felé az ún. GFX (Global Audio Effects) elemeket. Hozzá tartozik az equalizer, hangszóró/hangfal korekció, vagy pl. olyan kritikusan fontos dolog, mint a mélyhang kiemelés. A szerviz többek között ezeket az opciókat menti/tárolja/betölti munkamenetenként külön-külön. Ezt a szervizt leállítani nem lehet, ha egyszer elindítottuk, csak leltítani és a gépet újraindítani lehet. Ha viszont letiltjuk, az élményfokozás megszűnik.

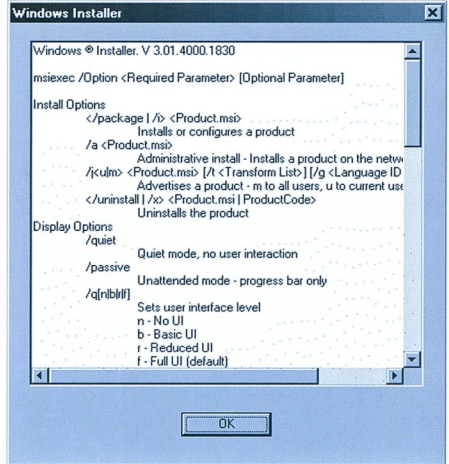
### Windows Installer (Windows Installer)

A szerviz rövid neve: *MsiServer*  
Az alkalmazás neve: *msiexec.exe /V*  
Függés: *Remote Procedure Call*  
Függésztés: -  
Porthasználat: *80*  
Alapértelmezett indítás: *kézi, leállítva*

A Windows Installer egy rendszeresen fókuszban álló komponens. Sokak által erősen vitatott a hatékonysága, ám tagadhatatlanul nehéz helyzetben van: minden gyári és külső alkalmazás rendszerbe állításakor képbén kell lennie, azért az ekkor begyűjtött információk alapján az alkalmazás módosítása, javítása vagy eltávolítása sikeres legyen.

A Windows Installer technológiának két fő, egymással szorosán együttműködő komponense van: az msiexec.exe (ami a telepítő szolgáltatás) és egy csomagfájl (.msi), amely gyakorlatilag egy relációs adatbázis. Az ebben tárolt információ alapján (konkrétan az msi.dll segít ezek kiolvasásában) történhet meg egy szabályos telepítés. E művelet során számtalan feladat van, az állományok merevlemezre másolásától kezdve, a registry módosításán vagy a párbeszédpanelek megjelenítésén illetve a felhasználó által variálható beállítás-

ok lekérdezésén át, egészen a társítási információk változtatásáig. És persze továbbra sem csak egy szimpla telepítőprogramról van szó, hanem extra feladatok is képhe kerülhetnek: ide tartozik pl. a telepítők bővítményeinek kezelése, vagy például a rollback, azaz a visszaállítás, amelyet számtalan területen használ a Windows, valamint testreszabható eszközt jelent a fejlesztők számára a különböző alkalmazások telepítőcsomagjainak elkészítésében.



### ■ És ez csak egy kis adag az msiexec.exe paramétereiből...

Ha a szerviz kézi indításúra van állítva, amikor egy telepítés kezdődne, automatikusra vált át. Amennyiben viszont le van tiltva, akkor sem telepíteni, sem eltávolítani, vagy telepítéseket javítani nem tudunk. Sőt, vannak olyan alkalmazások, amelyek a normál futásuk közben is használják ezt a szervizt, ergo ennek leltítása esetén nem is lesznek képesek futni.

Folytatjuk...

GÁL TAMÁS  
MCT, MCSE, MCSA, MVP  
gtamas@tjski.hu

### A cikkben szereplő URL-ek:

- [1] <http://tinyurl.com/6p6cy>
- [2] <http://www.microsoft.com/technet/scriptcenter/scripts/cls/ts/modify/default.msp>



# VPN karantén 2. rész

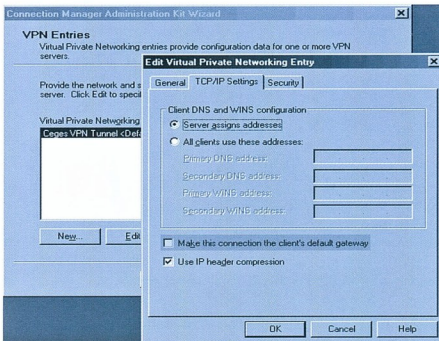
## WINDOWS SERVER 2003 + ISA SERVER 2004 KÖRNYEZETBEN • A CONNECTION MANAGER ADMINISTRATION KIT SZEREPE

Cikksorozatunk előző részében eljuttottunk a bevezetésen, a működés elméleti összefoglalásán keresztül az előkészítésig, azaz a Resource Kit passzoló elemeinek telepítéséig illetve a regisztrációs adatbázisban történő beállításáig. Az ISA 2004-en elkészítettük a speciális protokollt és szabályt, a végén engedélyeztük magát a karantént is. Ebben a részben megismerkedünk a Connection Manager Administration Kit-tel [amely nem csak VPN-Q eszköz, más távélérésű kapcsolatoknál is remekül használható], és elkészítjük a Connection Manager profilt, amelyben a VPN-Q kliens is benne lesz.

### A CMAK

A cikk szempontjából a CMAK (Csatlakozáskezelő felügyeleti csomag) csak a VPN-Q kapcsolat kliensoldali megvalósítása miatt érdekes, ám ez csak egy kis szegmense a „tudományán”. Alapfeladata az, hogy egyetlen, telepíthető állományt (.exe) létrehozva, az üzemeltető preparál egy távélérésű kapcsolatot, annak összes beállításával, majd valamilyen úton kiadja ezt a kliensnek (aki otthoni gépről, vagy „utazó” laptopról szeretne csatlakozni a vállalati hálózathoz), akinek csak el kell indítani az alkalmazást, és a távélérésű kapcsolat beavatkozás nélkül szépen feltelepül.

Ennek a módszernek számtalan előnye van. Azonkívül, hogy a felhasználónak nem kell kínlódnia a csatlakozás megteremtésével, az üzemeltető biztos lehet benne, hogy azok a beállítások jutnak érvényre amelyek beállított, ez biztonsági szempontból sem hátrány. Fontos még, hogy egységes környezettel rendelkeznek a „betárcsázó” ügyfelek, valamint az is, hogy ha valamilyen paraméter változik a szerver oldalon, akkor új csomagot kiosztva kliensoldalra, viszonylag egyszerűen átvihetők az új beállítások.

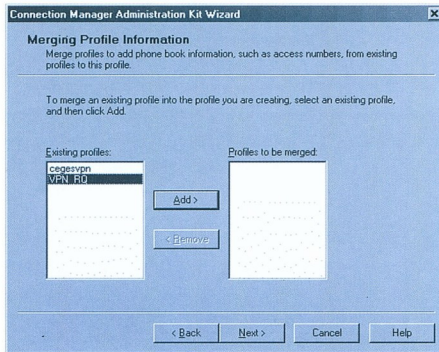


### ■ Egyszerűsíthető az életünk a távoli ájtájróra vonatkozó beállításal

Egy-egy profil részeként számtalan extra információt kiadhatunk az ügyfeleknek vagy olyan beállításokat is tehetünk, amelyeket egy manuálisan beállított kapcsolatnál nem. A teljesség igénye nélkül pár példát megemlítek:

- megjelentethetünk pl. a bejelentkezési panelen plusz (pl. támogatási) információkat, vagy csatolhatunk egyedi, akár általunk készített súgókat is (.hlp)
- megadhatunk egyetlen vagy több telefonszámot a csatlakozás létesítéséhez, de mellékelhetünk pl. egy szabályos telefonkönyvet is, amelynek az automatikus frissítését is kiköthetjük

- készíthetünk több telefonkönyvet, akár külön-külön a VPN illetve a telefonos csatlakozások számára, majd ezeket a különböző profilokhoz szabadon hozzá is rendelhetjük
- korrigálhatjuk például azt a bosszantó körülményt (lásd a mellékelt képen), hogy egy VPN kapcsolat kialakításakor az alapértelmezett átjáró mindig a VPN szerver lesz (othoni kapcsolat manuális kialakításakor rendszeresen jön a telefonhívás a felhasználóktól, hogy „vagy VPN van, vagy tudok netezni, mit csináljak???”)
- részletes biztonsági beállításokat is belementhetünk a profilba, pl. alpból letilthatjuk az állomány- és nyomtató megosztást, vagy beállíthatjuk a VPN protokoll kapcsolódási sorrendet,
- tehetünk a profilba a routing táblával vagy a proxy konfigurációval kapcsolatos kötelező érvényű beállításokat is,
- egyedi alkalmazások vagy szkriptek futtatására is van lehetőség (ezt ki is használjuk a karanténnál),
- fokozhatjuk a felhasználói élményt ☺, azaz egyedi (pl. céges) grafikákat illetve ikonokat is csatolhatunk a belépési panelhez, a telefonkönyvhöz,
- extra parancsokat fűzhetünk a kapcsolatot jelző státuszsorban lévő ikon helyi menüjéhez.



#### ■ Profilok gazdaságos összefűzése

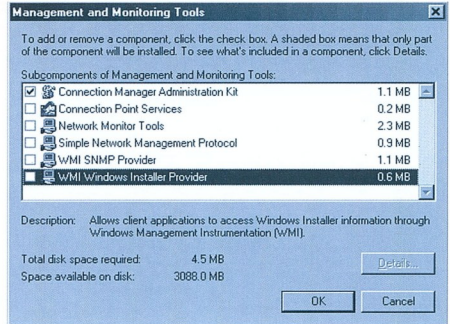
Egy-egy kapcsolatot egy-egy profilba ment a CMAK varázsló, ám ezeket a profilokat bármikor szerkeszthetjük, javíthatjuk, valamint a jövőbeni újabb profilkészítéskor ezeket ki is listázza alpból a varázsló, azért hogy pl. össze is fűzzük a bennük tárolt információkat, így spórolva az energiánkkal. De megtehetjük azt is, hogy az egész profilt integráljuk pl. egy IEAK (Internet Explorer Administration) csomagba. A profilok nem a Windows mappába kerülnek, hanem a program mappájába, konkrétan ide:

C:\Program Files\CMAK\Profiles

Egy profil mérete, spéci alkalmazások nélkül, a beállításokkal és néhány szkripttel kb. 500-700 KB (ebben maga a Csatlakozáskezelő legfrissebb verziója is benne van), ergo akár még egy floppyn is elvihető.

## CMAK telepítés

Alapértelmezés szerint a CMAK nincs feltelapítve, viszont a szokásos helyen, a „Management and Monitoring Tools” csop. portban megtaláljuk.



#### ■ Innen indulunk

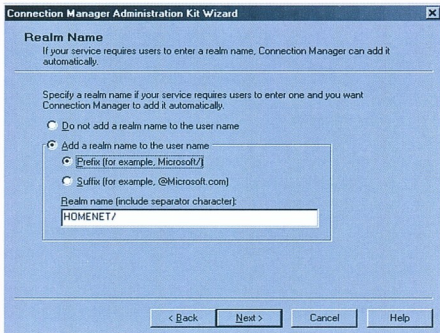
Érdemes itt megemlíteni még a CMAK alatti komponenst (lásd a mellékelt ábrán), a Connection Point Services-t, amelynek segítségével tartjuk majd karban a központi helyen a telefonkönyvet és ehhez fordulnak majd a kliensek - ha beállítottuk – a frissítésekért (ám ekkor még egy IIS-re is szükségünk lesz).

Telepítés után indítsuk el, majd kezdjük el a körülbelül 20 lépésből álló CMAK varázslóval a csomag kialakítását. Értelemszerűen egy új profilt fogunk készíteni. A „Realm Name” panelen megadható a tartománynév (így a felhasználónak még ezt sem kell), mint hitelesítési körzet, két fajta is (pre- és suffix), mivel elképzelhető, hogy Windows 9x/NT környezetből is lesz kapcsolódás.

Ezután jöhet a profilösszefűzés, amit most lépünk át, majd a telefonkönyv. Itt mindenképpen meg kell adnunk a VPN szerverünk nevét, címét. A következő panelen az „Edit...” gomb alatt viszont a fontos, korábban már ismertetett biztonsági beállítások, VPN kapcsolattípusok, TCP/IP paraméterek részletes beállítása történhet meg.

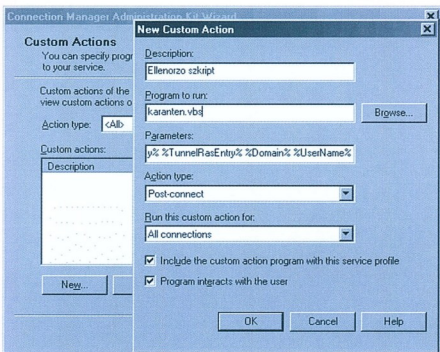
Most következhet a telefonkönyv kiválasztása, ha nincs ilyenünk, feltétlenül vegyük le a „Automatically download phone book updates” négyzetből a pipát, mert ha nem, akkor muszáj lesz kiválasztani a következő panelen.





## ■ A hitelesítési körzet megadása

A következő panelen az útválasztás, majd utána a proxy beállítások következnek. Eddig gyakorlatilag az összes beállítás a VPN karantén szempontjából érdektelen volt, ám most ez a helyzet változik. A következő ábrán is látható panelen készítsünk egy új bejegyzést az ellenőrző szkriptünk kijelölésére.



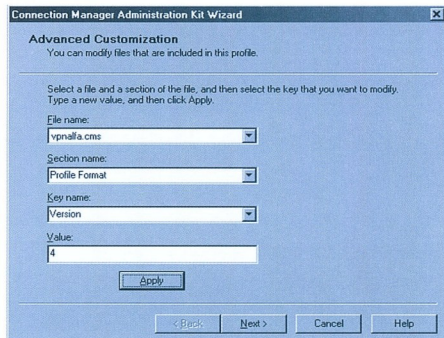
## ■ Az ellenőrző szkript futási körülményeinek beállítása

A százalékos paraméterek mindig azok az adott típusú ellenőrzést végző szkriptben szereplő értékek, amelyet majd Rqc.exe (a kliens) elküld az ISA-n futó Rqs.exe szervíznek, azért, hogy elkezdje a kapcsolat felépítését.

*(A Microsofttól is letölthető szkript csomagban [1] vagy a Step-By-Step Guide dokumentumban [2] egyaránt szerepelnek gyári ellenőrző szkriptek a cikkem első részében felsorolt célokra illetve az ISA 2004 MOC-ban is találhatóunk ilyet). Némi programozói tudással persze magunk is gyárthatunk saját ellenőrző szkripteket, ám egy biztos: nem lesz könnyű ☺.)*

Ezen a panelen be kell állítanunk még az ellenőrzés típusát, amely jelen esetben ún. „post-connect”, de ahogy a legördülő menüben is látható, többféle variáció is kiválasztható („Pre-dial”, „Pre-Connect”, „Disconnect”, „On-Error”, stb.). Ami

még fontos ezen a panelen, az „Include the custom action program with this profile” pont, amely bepipálása esetén az adott szkriptet is belemeni a csomagba a varázsló. Ezután a kapcsolódási folyamat különböző pontjain látható egyedi grafikák, ikonok kiválasztása jön, valamint a tálcá ikon helyi menü bővítésének lehetősége. Ezeket követi az egyedi sűgő, a támogatási információ megadása, majd a Connection Manager alkalmazás legfrissebb verziójának beagyazását tehetjük meg. A következő lépésben licenc információkat is adhatunk a csomaghoz, ezután viszont megint álljunk meg egy kicsit, mert ismét interakció jön, ez pedig az „Additional Files” opció. Ennél a pontnál kell ugyanis hozzáadnunk az Rqc.exe-t (Reskit mappa), amely ezzel a csomag része lesz. Abban az esetben ha pl. nyelvi válaszfájlok szeretnénk leküldeni a kliensre vagy egyedi alkalmazásokat akarunk használni, akkor ezeket is itt kell hozzátennünk a csomaghoz. A következő lépésben speciális, egyéni opciókat tehetünk hozzá a következő részben ismertetett konfigurációs állományokhoz. Innen ez kicsit meredek, de mindjárt látni fogjuk, hogy van átláthatóbb módszer is.



## ■ Az utolsó lépés

Most hagyjuk az extra beállításokat, lépünk tovább. Ekkor a varázsló legenerálja a csomagot, és ezzel készen is vagyunk, az adott profilban innentől megtalálható a kliensek használató .exe állomány.

## Plusz fájlok

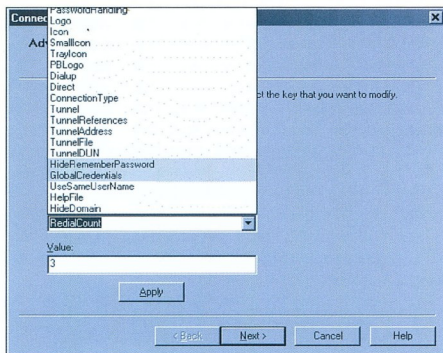
Amikor a Csatlakozáskezelővel létrehozunk egy profilt, a varázsló az .exe állomány mellett 4 darab, a megadott beállításokat tartalmazó fájlt is létrehoz a fent megjelölt helyen lévő profilmappákban:

- profil\_fájlnéve.cms
- profil\_fájlnéve.cmp
- profil\_fájlnéve.inf
- profil\_fájlnéve.sed

Ezekkel a további, kiegészítő beállításokat, a finomhangolást tehetjük meg. A legtöbb beállítást a .cms fájlban kell megadni, de az .inf és .cmp fájlok is szerkeszthetők. Mondok egy példát: a varázslóval nincs lehetőségünk arra, hogy beállítsuk a felhasználói jelszó automatikus elmentését. Ellenben ha a profilba tartozó .cms állományba, azon belül pedig a [Connection Manager] szakaszba beírjuk a

## HideRememberPassword

kulcsot, akkor ez láthatóvá és konfigurálhatóvá (0 = a jelölőnégyzet megjelenik a kliensen; 1 = nem jelenik meg) válik a Connection Manager utolsó paneljén (és persze a kliensen is).



Ha ezt valóban meg is tesszük, akkor célszerű felvenni továbbá a

## GlobalCredentials

kulcsot is, amely (csak XP és Windows Server 2003 esetén) meghatározza, hogy a kliensen el legyen-e mentve az első kapcsolódáskor a megadott felhasználói név és jelszó az összes felhasználó részére (0=mentse egynek; 1=mentse az összesnek). Ami nagyon fontos: ha változtatunk, akkor csak a változtatás után létrehozott profilokban fog érvényre jutni a változás. Ergo az első létrehozás, majd az ezekben az állományokban történő finomhangolás után újra le kell futtatnunk a varázslót, a profil frissítése miatt.

Ezzel végére is értünk a kissé terjedelmes, ám univerzálisan hasznosítható CMAK témakörnek, cikkem következő részében az elkészített csomag egyszerű publikálásáról, valamint a kliensoldali teendőkről ill. eseményekről lesz szó.

GÁL TAMÁS  
MCT, MCSE, MCSA, MVP  
gtamas@tjszki.hu

## A cikkben szereplő URL-ek:

- [1] <http://tinyurl.com/7275e>
- [2] <http://tinyurl.com/8n8b2>



reg. szám: 13-0830-04  
akkr. szám: 0082

1027 Bp., Csalogány u. 23. • Tel.: 457-6990 • Fax: 457-6990  
E-mail: training@controll.hu • Honlap: www.controll.hu  
Böngéssze honlapunkat!

**A WINDOWS SERVER 2003 ALAPÚ MCSA MINŐSÍTÉS! Kedvezményes tanfolyamsorozatunk 40% kedvezménnyel 4 teljes tanfolyam, 5 tananyaggal, 4 ingyenes vizsgával: csak 593 000 +ÁFA**

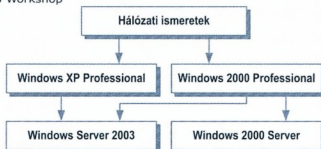
### Induló tanfolyamaink

- 2824 Implementing Microsoft Internet Security and Acceleration Server 2004 (ISA Server)
- 8036 Designing IT Platform Collaborative Applications with Microsoft SharePoint 2003 Workshop
- 2287 Managing Your Infrastructure Using Microsoft Operations Manager 2005
- 2400 Implementing and Managing Microsoft Exchange Server 2003
- 2596 Managing Microsoft Systems Management Server 2003
- 2733 Updating Your Database Administration Skills to Microsoft SQL Server 2005
- 2734 Updating Your Database Development Skills to Microsoft SQL Server 2005

### Windows 2003 hálózati rendszergazda

120 órában, délelőtt vagy délután - saját fejlesztésű magyar tananyaggal  
A modulok szabadon választhatók az ábra alapján!

1. Hálózati alapismeretek, - szept. 28-30.
2. Kliens operációs rendszer (XP vagy 2000) - okt. 17-25.
3. Szerver operációs rendszer (2003 vagy 2000) - nov. 14-18.





# ASP.NET 2.0 (Whidbey)

## Mi várható a 2005. évi ASP.NET-ben?

### CLIENT CALLBACK (ÜGYFÉLOLDALI VISSZAHÍVÁS)

Kevésbé reklámozott, de nagyon hasznos és trükkös új szolgáltatás a Client Callback. Segítségével anélkül változtathatjuk meg a böngészőbe betöltött lap tartamát, hogy a hagyományos postázásos módszerrel legeneráltatnánk újra a teljes lapot, így sokkal gyorsabb válaszidőket produkáló webalkalmazásokat írhatunk.

#### Alapvetés

Hagyományos módon a webalkalmazások a HTTP protokollnak megfelelően kérés-válasz módon működnek, azaz a böngésző kér valamilyen oldalt, válaszul kap egy (általában) html oldalt. Ezt megjeleníti, majd a felhasználó által kitöltött adatokat újra postázza a kiszolgálóra, ami erre újabb választ generál, stb. Jó ez, mert minimális tudást követel meg a böngészőtől, de nem jó a felhasználónak, mert minden egyes interakció újabb, meglehetősen lassú oldalgenerálást szül. Amióta a web elindult, igyekeznek csökkenteni a visszaposztázások számát, ennek két módja lehetséges: a letöltött adatokból táplálkozva programozzuk a böngészőben tárolt tartalmat, kihasználva a DHTML lehetőségeit, vagy a böngészőben futó kód „felnyúl” a kiszolgálóhoz, és az onnan kapott tartalmat beleszövi a már letöltött html tartalomba. Mindkét megoldás hátránya, hogy sokkal többet vár a böngészőben futó kód „felnyúl” a kiszolgálóhoz, és az onnan kapott tartalmat scriptből el kell tudni érni és módosítani.

DHTML programozás ma már minden modern böngészőben lehetséges, habár sokban kikapcsolható a scriptek futtatása, ilyenkor nincs mit tennünk, csak a hagyományos kérés-válasz modell fog működni.

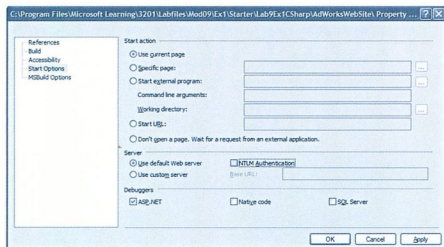
A második módszer még szigorúbb a böngészőkre nézve, mert szükséges valamilyen scriptből elérhető komponens, amivel HTTP vagy egyéb kapcsolatot építhetünk ki az adatforrásként használandó kiszolgálóval. Internet Explorer 5.5 fölött ez megoldott a böngészővel kapott MSXML programcsomag XMLHTTP komponense révén. Szerencsére Mozilla alatt is használható, így minden, a következőkben leírt programnak működnie kell Mozilla alatt is. Én Firefox 1.0.4-gyel teszteltem őket.

#### A Client Callback működése

Írjuk meg a Hello World-öt, mint klasszikus bevezetőpéldát. A cél az, hogy egy gombnyomásra a Hello World szöveget kiszolgálóoldali kód generálja, de azt ne postázással adjuk vissza, hanem Client Callbackkel. Először jöhet a böngészőben futó kód váza:

```
<script type="text/javascript">
// 
function helloButton_onclick() {
    content.innerHTML = 'alma';
}
// ]]&gt;
&lt;/script&gt;

&lt;/head&gt;
&lt;body&gt;
  &lt;form id="form1" runat="server"&gt;
    &lt;div&gt;
      &lt;input id="helloButton"
        type="button" value="Hello"
        onclick="return helloButton_onclick()" /&gt;
      &lt;br /&gt;
      &lt;div id="content"&gt;
      &lt;/div&gt;
    &lt;/div&gt;
  &lt;/form&gt;
&lt;/body&gt;</pre></div><div data-bbox="492 685 914 800" data-label="Text"><p>Az alma helyett szeretnénk a kiszolgáló által generált tartalmat beilleszteni. Mindjárt érdemes kipróbálni a fenti mégoly egyszerű scriptet Mozillában is, mert a böngészők objektummodellje sokban eltérhet egymástól. Az első kényelmetlenséggel azonnal szembesülünk: a Firefox jelszót vár tőlünk, annak ellenére, hogy az IE nem. Ennek oka, hogy a példát a B2-es Visual Studioban írom, és annak beépített web-szerverét használom.</p></div><div data-bbox="492 800 914 858" data-label="Text"><p>Az alapon Windows Authentikációt használ, ezért kért jelszót a Mozilla, míg az IE csendben elküldte a hitelesítési infókat. A kényelmesebb tesztelés kedvéért a projekt jellemzői között kikapcsoltam a hitelesítést.</p></div><div data-bbox="51 961 561 973" data-label="Page-Footer"><p>100% TECHNOLÓGIA ■ 0% MARKETING</p></div><div data-bbox="687 961 812 973" data-label="Page-Footer"><p>2005. 04.</p></div><div data-bbox="843 953 914 978" data-label="Page-Footer"><p>Microsoft<br/>TechNet</p></div><div data-bbox="937 961 963 973" data-label="Page-Footer"><p>25</p></div>
```



1. ábra: Windows hitelesítés kikapcsolása a VS.NET-ben

Azonnal szembesülünk vele, hogy a fenti kód nem megy Mozilla alatt, semmi nem történik a gomb megnyomására! A Tools/JavaScript Console ablakában semmi nyoma hibának. Egy tesztkód beszűrt `alert('alma')` működik, azaz legalább az eseménykezelőnk jó. A Mozilla DOM dokumentációt [1] tanulmányozva úgy tűnik, hogy az IE-ben megszokott rövidített objektumelérés nem megy a Firefoxban, hanem a `getElementById`-t kell használnunk. Igaz, ez a DOM szabványban [2] leírt működés, csak így jóval hosszabb kódot kapunk.

Ezzel a fenti eseménykezelőnk kódja így módosul:

```
document.getElementById('content').innerHTML = 'alma';
```

Itt az ideje átgondolni, mit kell tennünk kiszolgálóoldalon! A Client Callback technológia során az ügyféloldali script által küldött tartalmat egy `ICallbackEventHandler` implementáció képes lekezelni:

```
public interface ICallbackEventHandler {
    string RaiseCallbackEvent(string eventArgument);
}
```

Az `eventArgument` jön a böngészőtől, és a visszatérési értéként generált tartalom megy vissza ügyféloldalra. Az interfészt implementálhatja egy control, így saját magával társalgot az ügyféloldali és a kiszolgálóoldali része. Ezzel él is a `DetailsView`, `GridView` és `TreeView` is, ezeket megvizsgáljuk a cikk végén.

Ha nem egy vezérlőhöz kötődik a feldolgozás, akkor egyszerűbb magában a lapban implementálni az interfészt:

```
public partial class HelloPage :
    System.Web.UI.Page, ICallbackEventHandler
{
    public string RaiseCallbackEvent(
        string eventArgument)
    {
        return "Hello World!";
    } ...
}
```

Már csak egy lépés hiányzik: kellene generálni egy olyan ügyféloldali kódrészletet, amely képes kiváltani a `RaiseCallbackEvent` meghívását. Ezt nem kézzel írjuk

meg, hiszen nem ismerjük a Client Callback által használt script-függvények nevét, paramétereit, ezek az architektúra sajátjai. Segítségül vesszük viszont az erre a célra kialakított `GetCallbackEventReference` metódust:

```
string scriptDeclaration =
this.ClientScript.GetCallbackEventReference(
this, //1
"arg", //2
"HelloResponseArrived", //3
"kornyezet", //4
"HelloError", //5
True //6
);
```

A paraméterek jelentése:

1. Melyik vezérlő kezeli le az eseményt. Esetünkben a lap, ezért a `this`.
2. Hogyan hívjuk azt a paramétert az ügyféloldali függvényünkben, amiben a kiszolgálóra irányuló hívás paraméterét (string) adjuk át?
3. Ilyen néven kell létrehozunk egy ügyféloldali függvényt a lapban, amely majd a 2. és a 4. nevű paramétert várja paraméterül. Ebben kapjuk meg a kiszolgáló választát.
4. Egy paraméternév, amelyen a hívást kiváltó kód (esetünkben a click eseménykezelő) paramétert képes indirekten átpasszolni a hívás végét jelző függvénynek (3. paraméter). Arra jó, hogy így összeilleszthetjük a hívás megindítását a tényleges befejezéssel.
5. Hiba esetén ez az ügyféloldali metódus hívódjon meg.
6. Aszinkron módon történjen-e a visszahívás a kiszolgálóra? Miért ne?

A `GetCallbackEventReference` által generált script:

```
WebForm_DoCallback('__Page',arg,DisplayDateTime,
kornyezet,ErrorCallback,true)
```

Ez képes kiváltani a kiszolgálóoldali `RaiseCallbackEvent`-et. Az egyszerűbb kezelés érdekében ezt érdemes becsomagolni egy saját JavaScript függvénybe:

```
scriptDeclaration = string.Format(
@"function (1)(arg, kornyezet) {{ {0} }}",
scriptDeclaration, "Hello");
```

Az így nyert egyszerűsített függvény, ami kiváltja a visszahívást:

```
function Hello(arg, kornyezet)
{
    WebForm_DoCallback('__Page',arg,
    HelloResponseArrived,HelloError,true)
}
```

Ezt már csak bele kell ágyazni a lapba:

```
ClientScript.RegisterClientScriptBlock(
GetType(),
"HelloCallScript",
```



```
scriptDeclaration,
true);
```

Az új Hello függvényünkkel már könnyedén hívható a kiszolgálóoldal (nem adunk át semmilyen paramétert, a JavaScript nem haragszik érte):

```
function helloButton_onclick() {
    Hello();
}
```

A válasz pedig ide érkezik meg (aszinkron módon, egyszer csak):

```
function HelloResponseArrived(valasz, környezet) {
    document.getElementById('content').innerHTML =
        valasz;
}
```

Álljon itt a teljes kód, hogy egyben végig lehessen olvasni. A teljes lap forrása, fejlécek nélkül:

```
<script type="text/javascript">
function helloButton_onclick() {
    document.getElementById('content').innerHTML =
        '';
    Hello();
}

function HelloResponseArrived(valasz, környezet) {
    document.getElementById('content').innerHTML =
        valasz;
}

function HelloError() {
    alert('Hiba a kiszolgáló hívása közben.');
```

A háttérkód:

```
using Ez meg az;

public partial class HelloPage:
    System.Web.UI.Page, ICallbackEventHandler {
    public string RaiseCallbackEvent(
        string eventArgument) {
        return "Hello World!";
```

```
}
protected void Page_Load() {
    string scriptDeclaration =
        this.ClientScript.GetCallbackEventReference(
            this,
            "arg",
            "HelloResponseArrived",
            "környezet",
            "HelloError",
            true
        );
    scriptDeclaration = string.Format(
        @"function {1}(arg, környezet) {{{0}}}",
        scriptDeclaration, "Hello");
    statusLabel.Text = scriptDeclaration;
    ClientScript.RegisterClientScriptBlock(
        GetType(),
        "HelloCallScript",
        scriptDeclaration,
        true);
}
```

## Fejlett HTML tartalomgenerálás WebControlokkal

Az interneten keresgélve sok példát találni a Client Callback használatára, ám a legtöbb egy stringgé átalakított tömböt küldd vissza, és ebből generál a böngészőben futó script html tartalmat, pl. egy listát.

Alapvetően az nem tetszik ebben a megközelítésben, hogy a munka zömét ügyfélszoldalon kell leprogramozni, az pedig (szubjektív szempont) nekem nem barátságos környezet. Ha már egyszer vannak sokat tudó Server Controlok, mint pl. a DataGrid és társaik, akkor miért nem használjuk ki a html generálási képességeiket a Client Callbackek esetén is? A következő példa demonstrálja, hogy ez lehetséges. (Köszönet Kopinak és és Zokszigennek az ötletért).

A példa célja, hogy egy listából kiválasztva egy megyét, megjelenítsük az ahhoz tartozó települések listáját. Egy listában a megyék, másokban a települések, ám nem töltjük le az összes települést, hanem csak Client Callbackkel azt, amit éppen kiválasztottak (1. ábra).

A megoldáshoz kell egy megyék lista, amelyet SqlDataReader és egy asp:ListBox generál, a cikksorozat korábbi részeiben ismertetett adatkötés felhasználásával:

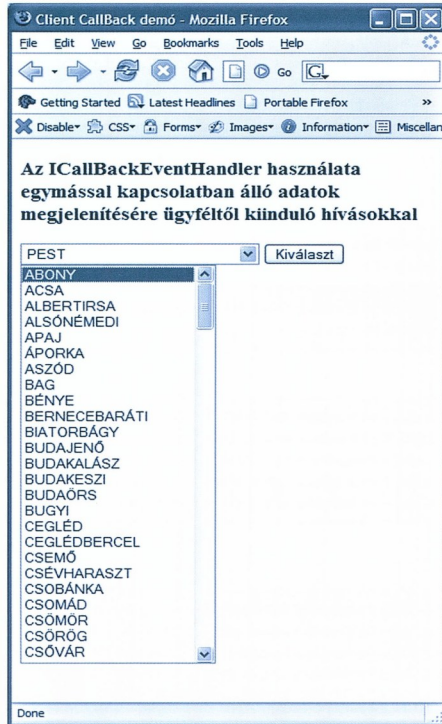
```
<asp:DropDownList ID="megyeLista" OnChange=
"TelepulesListaLetoltes(this.options[
this.selectedIndex].value, 'soci');"
runat="server" DataSourceID="megyeDataSource"
DataTextField="Nev"
DataValueField="ID">
</asp:DropDownList>
```

```
<asp:SqlDataSource ID="megyeDataSource"
runat="server" ConnectionString="<%$
ConnectionString:MOConnectionString %>"
EnableCaching="True"
SelectCommand=
```

```
"SELECT [ID], [Nev] FROM [Megye]
ORDER BY [Nev]">
</asp:SqlDataSource>
```

Az érdekes részek az előbbi lista `OnChange` eseményében történnek. Ez az ügyféloldali esemény meghívja a `TelepulesListaLetoltes` függvényt, ami már a kiszolgálóoldalról tölti le a kiválasztott megye településeit.

Ez a már látott módon a `GetCallbackEventReference` és társaival beregisztrált függvény.



2. ábra: Megyék-települések megjelenítése

A példa érdekessége kiszolgálóoldalon lapul:

```
int megyeID;

public string RaiseCallbackEvent(string
eventArgument)
{
    megyeID = int.Parse(eventArgument);

    string telepulesListaHtml =
        Cache[megyeID.ToString()] as string;

    if (telepulesListaHtml == null)
    {
        //Lista generálás és cache-elés
        Cache[megyeID.ToString()] = telepulesListaHtml
```

```
= GenerateTelepulesList();
}
return telepulesListaHtml;
}
```

A kapott megye paramétertől függően vagy elővesszük `Cache`-ből a településlistát, vagy generáljuk. A generálás a `razvas`, hisz most egy `Server Control` arra kell megkérni, hogy ne a szokásos `Response.OutputStream`-be generáljon tartalmat, hanem adja nekünk a kimenetét. Azaz lesz egy lista és egy adatforrás vezérlőnk a lapon, de a listát nem mutatjuk meg a lap betöltése során, hiszen ezt majd ügyféloldalról kezeljük. Viszont a VS 2005 tervezőit remekül használhatjuk, hiszen a vezérlőket a megszokott módon a lap tartalmazza:

```
<asp:ListBox ID="telepulesLista"
runat="server"
DataSourceID="telepulesDataSource"
DataTextField="Nev"
DataValueField="ID"
Rows="25" Visible="False">
</asp:ListBox>
```

Emlékeztetőül, `Visible="False"`, mert a lap szokásos kimenetében nem akarjuk, hogy megjelenjen.

Az adatforrás:

```
<asp:SqlDataSource ID="telepulesDataSource"
runat="server" ConnectionString="<%%$
ConnectionStrings:MOConnectionString %>"
SelectCommand="SELECT [Nev], [ID]
FROM [Telepules]
WHERE ([MegyeID] = @MegyeID)
ORDER BY [Nev]"
OnSelecting="telepulesDataSource_Selecting">
<SelectParameters>
<asp:Parameter Name="MegyeID"
Type="Int32" />
</SelectParameters>
</asp:SqlDataSource>
```

Az adatforrás `OnSelecting` eseményét le kell kezelnünk, mert át kell adnunk a szűrési feltételként, paraméterként a megye azonosítóját:

```
protected void telepulesDataSource_Selecting(
object sender, SqlDataSourceSelectingEventArgs e)
{
    e.Command.Parameters[0].Value = megyeID;
}
```

A megyeID-t már beállítottuk a felküldött megyéire a `RaiseCallbackEvent` metódusban (ld. pár blokkal korábban).

Már csak egy részlet hiányzik a teljes képhez: rá kell venni a listát, hogy egy stringbe generálja a tartalmat, ne a szokásos `Response`-ba:

```
private string GenerateTelepulesList()
{
    telepulesLista.Visible = true;
    telepulesLista.DataBind();
    telepulesLista.SelectedIndex = 0;
```



```

StringBuilder telepulesListaHtml =
    new StringBuilder();
HtmlTextWriter w =
    new HtmlTextWriter(
        new StringWriter(telepulesListaHtml));

    telepulesLista.RenderControl(w);
return telepulesListaHtml.ToString();
}

```

A `ListBox` kap egy általunk létrehozott `HtmlTextWriter`-t, és ahogyan a lap is szokta, megkérjük, generálja le a html tartalmát a `RenderControl` módszerrel. A példa cache-eli a tartalmat kiszolgálóoldalon, ez tovább javítható ügyféloldali cache-eléssel. Ha már egyszer letöltődött egy megyéhez tartozó településlista, miért ne tárolnánk el azt a böngészőben?

Egy lehetséges megoldást mutatok be a következő kód-blokkban:

```

var telepulesek = new Object();
function TelepulesekLejotttek(result, context) {
    telepulesek[context] = result;
    FillList(result);
}
function FillList(listContent) {
    document.getElementById(
        "telepulesListaPlaceholder").innerHTML =
        listContent;
}

//A korábbi eseménykezelő helyett
function MegyeListChanged() {
    var list =
        document.getElementById("megyeLista");
    var megyeID =
        list.options[list.selectedIndex].value;
    if (telepulesek[megyeID] != null) {
        FillList(telepulesek[megyeID]);
    }
    else {
        TelepulesListaLetoltes(
            megyeID, //arg
            megyeID); //context
    }
}

```

A megoldásban kihasználtam, hogy a `context` paraméterben átpasszoltam a megye azonosítóját a `MegyeListChanged`-ből a `TelepulesekLejotttek`-be (`GetCallbackEventReference` 4. paraméter).

A letöltött településlistát a `telepulesek` objektum `property`-jeiben tároltam el (egyfajta asszociatív tár vagy `Hastable` jellegű működés), `expando` property a hivatalos neve. Jól megfigyelhető a gyorsulás, ha IE-ben ráviszunk a fókuszra a megyelistára, majd a fel-le billentyűkkel mozogva a listában a települések lista először lomhán jön be, de miután egy-egy települést betárazott, nagyon gyors lesz.

### A hívások egyszerűsítése

A `GetCallbackEventReference` és társai logikus, de eléggé nehézkes felületet adnak a visszahívások kezeléséhez. Ha alkalmazásunk sok visszahívást használ, érdemes beburkol-

ni az előkészítő funkciókat egy `Server Control`-ba, ezzel megkönnyítve a szolgáltatás használatát.

Paul Glavich barátunk írt ehhez egy egyszerű vezérőt [3], annak használatát mutatom itt be. A forráskódot alaposan átvírva beleraktam a cikkhez mellékelt demóprojekthez is. Egyrészt az eredeti példa `ViewState`-ben tárolta a jellemzők adatait, ezt átvírtam `Control State`-re, így nem hiúlyd meg a vezérőt, ha kikapcsolják a `ViewState`-et.

Másrészt nagyon sok redundáns kód volt benne, azokat megszüntettem (a szerző szereti a kópi-pasztát).

A vezérő érdekessége, hogy nem csak kézzel váltható ki a visszahívás, mint az eddigi példákban, hanem periodikusan is, egy `timer`-re építve is. Ez automatikusan frissülő tartalomhoz jó lehet, de vigyázni kell vele, mert pár száz futó ügyfél komoly terhelést okozhat a kiszolgálónak.

Példa a vezérő használatára:

```

<cc1:AsyncClientScriptConnector
    ID="sc" runat="server"
    ClientSideCallbackMethod="NormalCallback"
    ClientSideErrorMessage="ErrorMessage"
    AutoRepeatInterval="1000"
    InitiateCallbackClientMethod="StartCallback"
    OnServerSideCallback="as_ServerSideCallback" />
<asp:TextBox ID="txtMsg" runat="server" />
<input id="button1" type="button" value="Manual
Call" onclick="return InvokeCallback()" />

```

Azaz paramétereznünk kell három ügyféloldali és egy kiszolgálóoldali függvénnyel, melyek felépítése hasonló a korábban leírtakhoz. Bővebben a példakódban [6].

### Objektumorientált, típusos adatátvitel

Az eddigi példában `string`-ek utaztak a két oldal között, mivel alapszintűen csak erre ad lehetőséget a technológia. Némi kézimunkával azonban összetett objektumokat is átvihetünk, így pl. egy kiszolgálóoldalon definiált `Employee` objektumot át tudunk vinni ügyféloldalra, ahol annak tagjaira szokásos objektumorientált módon `..tal` tudunk hivatkozni: `emp.LastName`.

A probléma azért nem egyszerű, mert az ügyféloldali JavaScript típusrendszere jelentősen egyszerűbb, mint a CLR típusrendszer, így hogyan lehet leképezni pl. egy `decimal`-t vagy egy `Guid`-ot JavaScript típusokra?

Persze fordítva is eljárhatnánk, és korlátozzuk magunkat kiszolgálóoldalon a JavaScript típusainak megfelelő CLR típusokra. Mindkét megoldás kompromisszumokkal terhes. Az első, proxi jellegű működést az Ajax.NET-ben figyelhetjük meg [4], ami nem a cikkben ismertetett módon, de `Client Callback` tesz lehetővé.

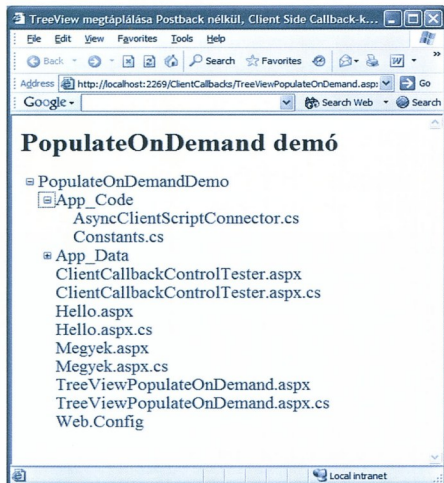
A két típusrendszer JavaScript alapú közös nevezőjét kereső írást szintén érdemes elolvasni, az az ASP.NET Callbackre épít [5]. Szerintem a Microsoft a következő verzióban fog valamilyen megoldást adni a problémára, addig a fenti címenek ötleteket láthatunk a saját implementációhoz.

### Client Callbacket használó vezérők:

#### TreeView

Ha már van ez az új szolgáltatás, miért ne építhetnénk rá az új vezérőt? A `TreeView` egy új, de mégis ismert vezérő. Az Internet Explorer WebControls már régóta tartalmaz egy fa implementációt, azt porolták le, írták át.

Nem céloim magát a vezérlőt részletesen bemutatni, ami számunkra most érdekes, az, hogy amikor a böngészőben a fa egy elágazására kattintanak, akkor kivátható egy kiszolgálóra irányuló visszahívás, amiben elő tudjuk állítani az éppen kinyitandó ág elemeit.



3. ábra. A fa ágainak kinyitásakor kiszolgálóoldalon fut le a tartalmat generáló kód

Indulásként egyetlen node, a gyöker lesz csak a fában:

```
<asp:TreeView ID="MyTree"
  PathSeparator="|"
  OnTreeNodePopulate="PopulateNode"
  ExpandDepth="1" runat="server">
  <Nodes>
    <asp:TreeNode Text="PopulateOnDemandDemo"
      PopulateOnDemand="true" />
  </Nodes>
</asp:TreeView>
```

Ha egy ágnak adatokra van szüksége, Client Callbackkel visszahívja a kiszolgáló `OnTreeNodePopulate` eseményben megadott metódusát.

A `PopulateNode` felépítése:

```
static readonly char[] _slashArray =
  new char[] { '/' };

protected void PopulateNode(Object source,
  TreeNodeEventArgs e)
{
  TreeNode node = e.Node;
  if (node.Value == "PopulateOnDemandDemo")
    node.Value = "~/";

  string rootDirectory =
    Request.MapPath("~/");
  Request.ApplicationPath, false);
  string fullPath =
    Request.MapPath(node.Value,
    Request.ApplicationPath, false);
```

```
//Egyszerű, de nem biztos,
//hogy tökéletes védelem,
//hogy ne lépjenek
//fel felsőbb könyvtárakba.
if (!fullPath.StartsWith(rootDirectory))
{
  return;
}

AppendDirectoryNodes(node, fullPath);
AppendFileNodes(node, fullPath);
}
```

Azaz kapunk egy referenciát egy, az ügyféloldali `Node`-ot reprezentáló objektumra (ami persze már egy kiszolgálóoldali típus), amiből megtudjuk, hová kattintottak. Ha nem akarják egyből a `windows\system32`-t olvasni, akkor legeneráljuk a könyvtárakat és a fájlokat tartalmazó `node`-okat:

```
private void AppendDirectoryNodes(TreeNode parent,
  string fullPath)
{
  string[] dirs =
    Directory.GetDirectories(fullPath);

  //Nézzük végig az összes könyvtárát
  foreach (string dir in dirs)
  {
    string virtualDir =
      parent.Value.TrimEnd(_slashArray) + "/" +
      Path.GetFileName(dir);

    TreeNode newNode =
      new TreeNode(
        Path.GetFileName(dir), virtualDir);

    //Ha még van alatta bejegyzés,
    //akkor ezt se töltjük fel előre
    if (Directory.GetFiles(dir).Length > 0
      || Directory.GetDirectories(dir).Length > 0)
    {
      newNode.PopulateOnDemand = true;
    }
    parent.ChildNodes.Add(newNode);
  }
}
```

A dinamikusan létrehozott könyvtárak is `PopulateOnDemand = true`-ra vannak állítva, így őket is ki lehet majd nyitogatni. A fájlok listázásában egyedül a fájlnevekkel való bűvészkedés okozott némi fejtörést:

```
private void AppendFileNodes(TreeNode parent,
  string fullPath)
{
  string[] files = Directory.GetFiles(fullPath);
  foreach (string file in files)
  {
    TreeNode newNode = new TreeNode(
      Path.GetFileName(file),
      Path.GetFileName(file));

    //Tudom, tudom, csúnya...
    newNode.NavigateUrl =
      Request.ApplicationPath + '/' +
      file.Replace(
        Request.PhysicalApplicationPath, "");
    Replace('\\', '/');
    parent.ChildNodes.Add(newNode);
  }
}
```



## Client Callback használó vezérlők: GridView, DetailsView

A GridView képes Client Callbackkel rendezetni valamely oszlopra kattintva a táblázat tartalmát, illetve képes lapozni, anélkül, hogy a teljes oldal újratöltődne. Ehhez csak annyit kell tennünk, hogy az EnableSortingAndPagingCallbacks jellemzőt true-ra kell állítani. A DetailsView lapozni tudna callbackekkel, ha az EnablePagingCallbacks-szel azt engedélyezzük. Sajnos egyik vezérlőben sem működnek még a visszahívások a Beta2-ben (nekem), ezért ezt a témát egyelőre elnapoljuk.

## Példakódok

A cikkben kidolgozott példakódok jóval hosszabbak és jobban kommentezettek, mint ahogyan itt közzöltük, ezért érdemes megnézni őket teljes „pompájukban” a [6] címen.

## Zárszó

Érdekes új szolgáltatás a Client Callback, amelyen még látszik az újszülött kor, de ennek ellenére újabb ravasz fegyvert ad a kezünkbe kényelmesebb Webalkalmazások írásához. Ahogyan egyre többen használják a 2.0-s ASP.NET-et, úgy

valószínűleg sokféle, magasabb szintű megoldás fog kialakulni, amelyek egyszerűbbé teszik a visszahívások elven működő alkalmazások írását. Ám, mint minden új, kevésbé kitaposott út, izgalmas gondolkodási lehetőséget biztosít a kreatív elmék számára eddig nem látott alkalmazások kifejlesztésére. Sok sikert ehhez a Kedves Olvasónak!

SOCZO ZSOLT

zsolt.soczo@netacademia.net

A szerző a NetAcademia vezető fejlesztőoktatója  
ASP.NET MVP, MCSE, MCSD, MCDBA, MCT

## A cikkben szereplő URL-ek:

- [1] [www.mozilla.org/docs/dom/](http://www.mozilla.org/docs/dom/)
- [2] [www.w3.org/DDM/](http://www.w3.org/DDM/)
- [3] [weblogs.asp.net/pglavich/archive/2005/05/11/406348.aspx](http://weblogs.asp.net/pglavich/archive/2005/05/11/406348.aspx)
- [4] [weblogs.asp.net/mschwarz/](http://weblogs.asp.net/mschwarz/)
- [5] [weblogs.asp.net/bleroy/archive/2005/05/19/407539.aspx](http://weblogs.asp.net/bleroy/archive/2005/05/19/407539.aspx)
- [6] [netacademia.net/tudastar/articlepage.aspx?upid=6658](http://netacademia.net/tudastar/articlepage.aspx?upid=6658)

# Microsoft Worldwide Partner Conference 2005

„Az elkövetkező két évben valamennyi termékvonalunkon fogunk új, integrált szoftvermegoldásokat szállítani – olyan megoldásokat, amelyek páratlan üzleti értéket és jelentős előnyöket teremtenek partnereink számára világszerte” - mondta Steve Ballmer, a Microsoft vezérigazgatója a vállalat nemzetközi partnerkonferenciáján.

A konferencián első alkalommal mutatták be nyilvánosan azon űrlapfunkciókat, amelyek a Microsoft Office Rendszer jelenleg „Office 12” névvel jelölt következő generációjában fognak szerepelni. A Microsoft bemutatta az RTC eszközkészletet, amely a Microsoft Visual Studio projektekben felhasználható vizuális vezérlőket tartalmaz. Ezek használatával az alkalmazásfejlesztő partnerek egyszerűen be tudják építeni a jelenléttel, azonnali üzenetküldéssel és hívásvezérléssel kapcsolatos funkciókat a saját üzleti alkalmazásaikba.

A Microsoft Windows következő jelentős verziója egy újfajta számítástechnika alapjait teremti meg. A Longhorn a biztonság és megbízhatóság alapfunkcióitól kezdve, a látványos, új megjelenésig minden területen jelentős előrelépéseket hoz. A Microsoft arra ösztönzi a fejlesztőket és a független szoftvergyártókat, hogy teljes mértékben használják ki a Longhornhoz kapcsolódó, jelenleg WinFX néven ismert új programozási modellt, amely kiterjeszti a .NET-keretrendszert, és újszerű alkalmazások létrehozására ad lehetőségeket az iparági partnerek számára. A Longhorn első bétaverziója idén nyáron várható.

A Microsoft piacra dobja a Windows Small Business Server 2003 R2 verzióját, amely a Windows Server 2003 R2 kibocsátását követő 60-90 napon belül kerül gyártásba. A kisvállalkozásokat célzó új termékek és szolgáltatások többek között a következők: SQL Server 2005 Workgroup Edition, javítás- és frissítés-kezeléssel, illetve megnövelt méretű postafiókokkal (75 GB).

A Microsoft System Center Data Protection Manager (DPM) a Microsoft első lépését jelenti a lemez alapú biztonsági mentési és helyreállítási területen. A DPM csomag egy DPM kiszolgálói és három kezelési licenccel fog tartalmazni. A Data Protection Manager termék a Microsoft Partner Program keretében a Speciális Infrastruktúra kompetencia Rendszerfelügyeletébe lesz integrálva, ilyen módon a partnerek számára többek között projektkezelőkből, online tanfolyamokból, vizsgákból és minősítésekkel álló átfogó eszközkészlet és oktatás válik hozzáférhetővé.

Az SQL Server 2005, a Visual Studio 2005 és a Biz Talk Server 2006 ez év novemberében fognak megjelenni.

<http://www.microsoft.com/partner/events/wspartnerconference>

# IT folyamatok javítása

Előző cikkemben egy, a hazai IT cégek körében végzett felmérést ismertettem. Most ahhoz kapcsolódóan mutatok be néhány, a folyamatok javítására irányuló ötletet, praktikát. A felmérés számos olyan problémára mutatott rá, amelyek megoldása alapos körütekintést igényel. Azt tapasztalhatjuk ugyanis, hogy minden mindennel összefügg: ha a vállalati kultúrát vagy infrastruktúrát egy helyen megváltoztatjuk, az kihatással lesz a többi tényezőre is.

**A**vállalat feladatainak elvégzéséhez bizonyos létszámra mindenképp szükség van. Ezt a létszámot gondos mérlegelés alapján határozhatjuk meg: lehet, hogy a mostani állományhoz új embereket kell még felvenni, de az is elképzelhető, hogy bizonyos átszervezések után létszámleépítésre lesz lehetőség (szükség). A szervezeti hierarchia és kultúra megváltoztatása azonban rendkívüli körütekintést igényel, hiszen egyrészt ettől függ a későbbi munkavégzés hatékonysága, másrészt enélkül rendkívüli ellenállásba ütközhetünk munkatársaink részéről.

A szervezeti felépítés átgondolásánál, újratervezésénél arra kell ügyelnünk, hogy sem a túl „lapos”, sem a túl „me-redek” hierarchia nem eredményez hatékony munkavégzést – az optimumot kell megtalálnunk úgy, hogy az a lehető legkevesebb „fájdalommal” járjon a meglévő szervezeten belül.

## Projektek, folyamatok szervezése

Napjainkban gyakorlatilag mindenki projekt-környezetben él: munkánk folyamatosan specifikációkból, határidőkből, éjszakázásokból, költségkeretből és minőségi kritériumokból áll. Folyamataink – a felmérés tanulsága szerint – azonban nagyrészt ad hoc módon zajlanak, és még ha van is felső szervezettség, az alsó szint – közvetlenül – ebből nem sokat érez. Projektjeinket, és az azokhoz kapcsolódó folyamatainkat gyakran fejtelenség jellemzi, és ez már önmagában is negatív élt ad a fogalomnak.

Vezetőként azonban sokat tehetünk azért, hogy folyamatainkat jól szervezzetek, áttekinthetőek, és mindenki számára érthetőek, kiszámíthatóak legyenek.

Először is ki kell választanunk egy olyan eszközt, amely automatizálja, egyszerűsíti folyamatkezelésünket. A Microsoft Project segítségével nemcsak tervezhetjük folyamatainkat,

de nyomon is követhetjük őket, sőt a csoportmunkát is megszervezhetjük.

Sokak számára ismerősek az úgynevezett Gantt-diagramok, melyek a projekt tevékenységeinek időtartamát és egymás-

ra hatását szemléltetik. Azt azonban már kevesebben tudják, hogy az MS Project segítségével különféle számításokat, összehasonlításokat és kimutatásokat is készíthetünk, összevetve a tervek a jelenlegi helyzettel, sőt a Project Server segítségével még a csoportmunkát is összehangolhatjuk.

Fontos emellett, hogy ne csak a technikai hátteret teremtsük meg, hanem a napi gyakorlatba is feltétlenül építsük be annak használatát, illetve az ezzel járó plusztevékenységeket.

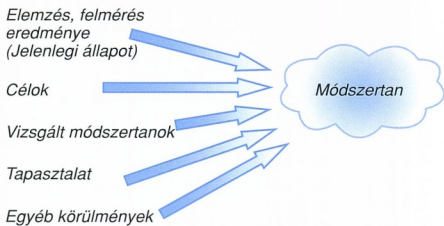
## Fejlesztési módszertanok

A projekt-folyamatok rendbetétele mellett a szakmai, szoftverfejlesztési tevékenységek szabályozása, meghatározott keretek közé foglalása is rendkívül fontos lépés. A nagyobb, jól szervezett vállalatok többsége kidolgozott olyan módszertanokat, melyek náluk beváltak, és bizonyítottan biztosítják a megfelelő rendet. A Microsoft szoftverfejlesztési keretrendszere az MSF (Microsoft Solutions Framework), amelynek következő verziója a Visual Studio Team Systems-szel (VSTS) egyidőben jelenik majd meg. Az MSF4 elsősorban abban jelent majd újdonságot, hogy a korábbi verziók „nagy” (MSF Formal) módszertana mellett az MSF Agile a kisebb terjedelmű, egyszerűbb projektekhez nyújt majd segítséget. Emellett a VSTS az informatikai infrastruktúrát is biztosítja az MSF4 megvalósításához, amely újabb nagy segítséget jelenthet majd.

Természetesen a „kész” módszertanokat nem elég egy az egyben átvenni, azokat igazítani kell saját igényeinkhez, követelményeinkhez és lehetőségeinkhez.

**Napjainkban gyakorlatilag mindenki projekt-környezetben él, folyamataink azonban nagyrészt még mindig ad hoc módon zajlanak, projektjeinket és folyamatainkat gyakran fejtelenség jellemzi.**





#### ■ Saját módszertan kialakítása

### Szabványok, ajánlások

A projektek és fejlesztési folyamatok rendbetétele mellett a vállalati kultúra is jelentős átalakításra szorulhat.

Az IT projektek során a tervezés mindig kritikus kérdés, hiszen a projekt sikere vagy sikertelensége nagyrészt már itt eldől. Célszerű egységesíteni nemcsak a tervezéshez használt eszközöket (pl. MS Visio), de az egyes technikákat, az alkalmazott Design Pattern-ek körét, tárolási módját (DP-tár) is.

A tervezés mellett magára a kódolásra is hasznos lehet olyan ajánlásokat kidolgozni, amelyek a megfelelő keretek között tartják fejlesztőink munkáját, ám nem jelentenek betarthatatlan korlátokat számukra. Csak az olyan szabályoknak van ugyanis értelme, amelyeket be lehet tartani és tartatni. Ebben a szabványban az alábbi dolgokra érdemes kitérni:

- elnevezések (névterek, osztályok, interfészek, enumok, események és eseménykezelők, tulajdonságok, eljárások, paraméterek és tagváltozók, stb.)
- felhasználói felület elemeire vonatkozó konvenciók
- adatbázis konvenciók (kulcsszavak, táblák, nézetek, indexek és kulcsok elnevezései, tárolt eljárások formátuma, stb.)
- forráskódok formátuma, olvashatósága, file-ok szerkezete
- hibakezelés

Emellett a verziókezelésre, a build folyamatokra és egyéb fejlesztői tevékenységekre is rögzítsük a megfelelő szabályokat. Végül ezeket a szabályozással kapcsolatos dokumentációkat olyan helyen kell tárolni, ahol bárki hozzáférhet, és megoldott a hozzászólások, módosítási javaslatok megfelelő kezelése is – erre jelenthet megoldást egy SharePoint portáloldal, amelyet a központi vállalati portálhoz integrálunk.

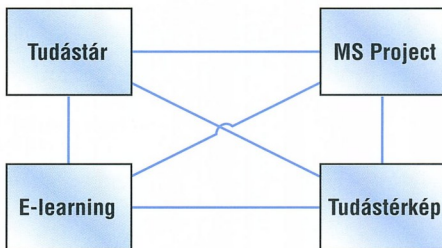
### Képzések

A vállalat életében a képzésekre fordított idő és energia különösen kritikus kérdés, sokan ugyanis sajnos csak a szükséges ráfordításokat látják, azt nem veszik figyelembe, hogy mindez egyfajta befektetés. Már a mindennapi működéshez is elengedhetetlen a folyamatok továbbképzés, hiszen lépést kell tartani a folyamatos technológiai változással. A képzési kultúra megváltozása mellett figyelni kell a megfelelő technológiai-informatikai háttérre is. Ma már számos

olyan megoldás áll rendelkezésünkre, amelyek segítségével nem csak a hagyományos oktatási módszerek használhatjuk céljaink eléréséhez.

Először is érdemes egyfajta belső, céges tudástárat létrehozni, ahol a munkatársak strukturált módon tárolhatják a különféle dokumentumokat, leírásokat, illetve hasznos kódrészleteket. Természetesen elképzelhető, hogy ezek a funkciók több rendszerben helyezkednek el, amelyeket integrálva, egy felületen (is) használunk.

Dokumentumok, leírások, illetve projektekhez tartozó egyéb hasznos dolgok tárolására, rendszerezésére hatékony megoldást kínál a SharePoint Server, amely emellett még az integrációs felület szerepét is elláthatja. A wabpart-ok segítségével ugyanis funkcionalitása tetszőlegesen bővíthető, egyéb rendszerekkel integrálható. Így például innen is nyomon követhetjük projektjeink állását, linkeket helyezhetünk el különféle rendszereinkre, egy teljes vállalati portált kiépítve ezáltal.



#### ■ A képességekkel kapcsolatos háttér-rendszer együttműködése

Ezen kívül hasznos lehet egy e-learning felület létrehozása is.

Egy olyan háttér-adatbázisra („tudástérkép”) is szükség lehet, amelyben azt tároljuk, hogy kinek milyen képesítései, végzettségei vannak, milyen tanfolyamokon vett részt, esetleg mik a jövőbeli tervei, stb. Természetesen ennek naprakészségét is biztosítani kell, és a hagyományos CV-nézet mellett kereső- és összehasonlító felületet is létre kell hoznunk. Amennyiben ezt is integráljuk portálunkhoz, lehetőségünk nyílik olyan összetett feladatok megoldására is, mint például a tudástár bejegyzéseinek személyhez kötése, vagy az egyes projektekhez a megfelelő szakértelem összehívogatása.

Végül, de nem utolsósorban meg kell említeni a személyes képzések jelentőségét is: hiába minden automatizmus és háttér-infrastruktúra, a személyes kapcsolatokat és konzultációt esetenként semmi nem pótolhatja. Különösen igaz ez abban az esetben, ha gyakorlati tudást, tapasztalatot akarunk átadni.

A hagyományos előadások helyett célszerű lehet kihasználni azokat az időreszeket, amelyek az egyes projektek között hú-

**A vállalat életében a képességekre fordított idő és energia különösen kritikus kérdés, sokan ugyanis csak a szükséges ráfordításokat látják, azt nem veszik figyelembe, hogy mindez egyfajta befektetés.**

zódnak: ilyenkor az emberek kevésbé (vagy egyáltalán nem) terheltek, nagyobb kapacitásuk van a tanulásra, és figyelmük is koncentráltabb lehet. Arról se feledkezzünk meg, hogy a gyakorlat-orientált szemléletű fejlesztők sokkal szívesebben és hatékonyabban sajátítják el azokat az ismereteket, amelyekről látják, pontosan mire is használhatók. Ezért ha csak tehetjük, a tudás átadása mellett olyan pilot-projekteket is indítsunk, amelyek elsődleges célja az ismeretszerzés, ám mint egy „melléktermékként” hasznos termékeket (pl. belső használatú szoftvereket vagy komponenseket) is létrehozunk.

### Kommunikáció

A fentiek megvalósításához elengedhetetlen, hogy az érdekelte munkatársakban a megfelelő elkötelezettséget alakítsuk

ki. Ehhez pedig hatékony kommunikációs csatornákra és vezetésre, valamint megfelelő tervezésre van szükség. A vezetői hozzáállást, illetve a szervezetben rejlő kommunikációs lehetőségeket igen nehéz befolyásolni – de nem lehetetlen. Nagyon sokat jelenthet egy jó háttér-rendszer, amelyet szinte öröm használni. Ehhez azonban alapos tervezésre van szükség – mind szakmailag, mind tartalmilag, mind pedig ergonomiaiilag.

Sok sikert hozzá!

MOLNÁR ÁGNES  
tanácsadó  
molnar.agnes@clwware.hu

## TechEd Europe 2005

A tizenkettedik alkalommal, Amszterdamban rendezett Tech Ed 2005 Europe konferencia a Microsoft legjelentősebb technológiai fóruma, melyen az idén is kiemelten fontos bejelentések hangzottak el.

**A**júliusi bejelentések közül kiemelt figyelmet övezte az Active Directory technológiával kapcsolatos új kezdeményezéseket, köztük az Active Directory együttműködési programot, melynek célja, hogy a partnerek hozzáférhessenek az Active Directory rendszerbe beépített széleskörű együttműködési képességekhez. A Microsoft elérhetővé tett továbbá egy új, külső fejlesztésű OpenLDAP felügyeleti ügynökprogramot a Microsoft Identity Integration Server 2003 termékhez, illetve a Microsoft identitáskezelési stratégiáját kiteljesítő Active Directory Federation Services (ADFS) felügyeleti szolgáltatás új, külső támogatását.

További információ: [http://www.microsoft.com/emea/presscentre/PressRelease.aspx?i=MicrosoftCreatesProgrammes\\_050705](http://www.microsoft.com/emea/presscentre/PressRelease.aspx?i=MicrosoftCreatesProgrammes_050705)

A Microsoft bejelentette továbbá, hogy az Office 12 rendszer a szabványos XML technológia használatát fogja támogatni (lásd még ugyanerről cikkünket a 3. oldalon), így az Office 12 rendszerben a dokumentumok, táblázatok vagy bemutatók létrehozásakor és mentésekor az új, XML alapú fájlformátum lesz alapértelmezett. Az Office termékekben használt szabványos XML technológia támogatásával kapcsolatban további tudnivalók az alábbi címeken találhatóak:

<http://www.microsoft.com/presspass/features/2005/jun05/06-01XMLFileFormat.mspx>

<http://www.microsoft.com/presspass/press/2005/jun05/06-01OfficeXMLFormatPPR.mspx>

Az SQL Server 2005, a Visual Studio 2005 és BizTalk Server 2006 megjelenése november 7-én esedékes, San Franciscóban, melyet több mint 50 országban a nemzeti ver-

zió megjelenése fog követni. Az új verziók bejelentése mérőföldkönek számít, hiszen ezek révén a Microsoft platformot biztosító egymáshoz kapcsolódó rendszerek következő generációjának, a fejlesztők számára pedig olyan infrastruktúrát terem, amely páratlan termelési hatékonyságot kínál, lehetővé teszi a létfenntartás alkalmazások hatékony kezelését, és átláthatóbbá az üzleti döntések meghozatalának folyamatát. A termék megjelenésekkel, a technológiai továbbfejlesztésekkel és a partnerek támogatásával kapcsolatban további tudnivalók:

<http://www.microsoft.com/emea/presscentre/techEd/default.mspx>

A Microsoft álláspontja szerint egy szervezete sikerességének kulcsa magukban az emberekben rejlik, ezért a vállalat továbbra is teljes erővel támogatja a DSI-t, (Dinamikus Rendszer Kezdeményezés, Dynamic Systems Initiative), mely keretet ad a vállalati és informatikai szabályzatok betartásához, emellett csökkenti az informatika általános felügyeleti költségeit. A DSI az egész iparágra kiterjedő kezdeményezés, amely az informatikai csoportok a rendszerek teljes életciklusát felölelő, a fejlesztők, informatikai szakemberek és infómunkások csoportjait is magába foglaló modelleken keresztül rögzíti, és adja át az információkat. A „Rendszerdefiníciós Modell” (System Definition Model – SDM) – a DSI egyik legfontosabb technológiája – is szerepelni fog a Visual Studio 2005 Team System rendszerben, illetve a későbbiekben megjelenő Microsoft System Center felügyeleti megoldásokban, ezzel lehetővé válik az üzemeltetési vezetők és alkalmazás-tervezők hatékony együttműködése, és a rendszerek tervezése.



# Ami a hivatalos Microsoft tanfolyamokból kimaradt...

## ACTIVE DIRECTORY – WEBES REGISZTRÁCIÓ (2. RÉSZ)

Folytatjuk weblapunk készítését, melynek segítségével a felhasználók maguk kérvényezhetik hozzáférési szándékukat a rendszerünkhöz, intranetünkhöz. A már elkészített önregisztráció alapján, az automatikusan létrehozott felhasználói fiókokat tovább boncoljuk, és elkészítjük az e-mail alapú értesítéseket, továbbá megnézzük, hogyan „élesedik”, és vándorol a létrehozott felhasználói fiók egyik OU-ból a másikba. De először megnézzük, miként lehet a „shell”-t IE-re cserélni az önregisztráció első lépcsőjéhez.

**M**ielőtt belevágnánk a „shell” csere középsébe, elevenítsük fel, hogy hol is tartunk: Felépítettünk egy virtuális várost annak érdekében, hogy megfelelő méretben, és felhasználói környezetben tudjuk elképzelni a webes regisztráció hangsúlyát. Van tehát 100 végpontunk, 500 munkaállomásunk, 3000 felhasználónk, és egy elcsigázott rendszergazdánk (aki a webes regisztráció végállomásán ül, és pipálgat). A technológia ISDN, de legfeljebb KábelTV alapokon nyugszik. Webes technológián készítetünk regisztrációs űrlapot, amelynek segítségével az igényléseket elkészíthetik az önjelölt felhasználók. Létrehoztunk egy AD struktúrát, melyben helyet kaptak az intézmények, és azon belül a jogosultsági (vagy inkább az állapot) OU-k. Elkészítettük a biztonsági csoportokat is, és behangoltuk a web.config fájlt a későbbi kényelmes módosíthatóság érdekében.

Persze az általam ajánlott megoldás csak egy a lehetséges megoldások közül. A felvázolt OU-k természetesen lehetnének teljesen más hierarchiában is, az nem változtatná meg a szerepüket. Sőt, ezt a szerepet át lehet bátran ruházni csoporttagságokra, vagy akár ritkán használt, és emiatt általában üresen hagyott AD jellemzőkre. Bátorabbak akár „schema” módosítást is tehetnének a „nagyon elegáns” megoldás érdekében. Az általam ajánlott OU rendszerezésnek az a lényege, hogy ha tehetjük, ne használjunk más háttértartat az AD-n kívül. Tehát, ha egy felhasználót szeretnénk létrehozni a rendszerünkbe, egyszerűen hozzuk létre (közvetlenül, vagy webform-on keresztül), és ha (biztonsági, vagy egyéb okokból) aggodalmaink vannak a használatával kap-

csolatban, akkor tiltott állapotban keltsük életre, majd hagyjuk valakire az engedélyezését. Ha mindez több lépésben szeretnénk megtenni, akkor valahol tároljuk a felhasználó objektumunk állapotinformációit úgy, hogy az mindig kéznél legyen (pl. az objektum egy tulajdonságaként vagy az AD-ben elfoglalt helyeként).

Most térjünk vissza a kezdeti lépéshez, az igényléshez magához. Az alapkérdés az, hogy miként tud a kérelmező felhasználó hozzáférni a rendszerünkhöz igénylés céljából, ha még nincs is hozzáférési engedélye. Mint írtam már, több jó megoldást is el tudok képzelni: Kinevezünk egy adott munkaállomást intézményenként, és felügyelet mellett folyik az igénylés. Esetleg kirakjuk egy nyilvános weblapra az igénylési webform-ot (nincs ezzel gond, hisz jóváhagyáshoz kötött a folyamat). Mindezek ellenére most mi a „shell” cseréléséhez folyamodunk. Ennek számtalan praktikus oka lehet annak ellenére, hogy igen ritkán használt eljárás. Pl. a beléptető felhasználó bejelentkezési adatai teljesen nyilvánosak lehetnek. Továbbá nincs szükség kijelölni egy adott munkaállomást (munkaállomásokat) a feladatra, a rendszerünk bármely munkaállomását használhatjuk regisztrációra. De nem elhanyagolható az az érv sem, hogy a „shell” csere által nincs lehetőség semmilyen Windows alapú „nyulkapiszkára”. Nincs „start” menünk, nincs eszköztárunk, és semmi egyebünk, amit a „desktop” biztosít nekünk alapesetben. Csak az IE van teljes képernyőn, és ami abban elérhető. Ha kilépünk a webform-ból, azaz bezárjuk az IE-t, együttál ki is jelentkeztünk a munkaállomásunkból.

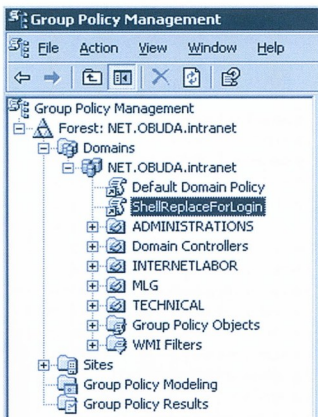
## „Shell” csere IE-re

Rendszerünk felépítése egy központi szerverstruktúrára épül, ezért kézenfekvő a „shell” cseréléséhez a GPO használatát. Az is fontos, hogy a GPO ne minden felhasználóra legyen érvényes, hanem csak azokra a megjelölt beléptető felhasználókra, akik kifejezetten erre a célra lettek létrehozva. Célszerű tehát minden intézményi OU alá létrehozni egy beléptető felhasználót, és olyan néven létrehozni őket, ami utal az intézményeikre is (előző számban már tárgyaltuk mibenlétüket). Magától értetődő, hogy csoportba rendezve könnyebb a kezelhetőségük. Ezen csoport lesz a „Kérelmezők” csoportja, melyet az előző számban már létrehoztunk. Bezárult egy kör.

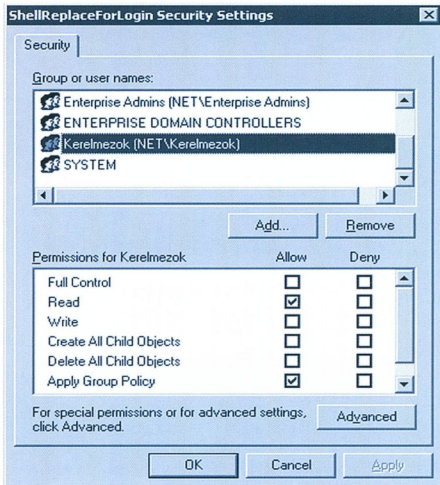
Kezdünk neki a GPO-nak. A tartományunk tetején (legfelső szintjén) hozunk létre egy új GPO-t, mondjuk „ShellReplaceForLogin” néven.

Link Order	GPO	Enforced	Link Enabled	GPO Status	WMI Filter
1	Default Domain Policy	No	Yes	Enabled	None
2	ShellReplaceForLogin	No	Yes	Enabled	None

Állítsuk be a jogosultsági szintjét a megbeszéltek szerint. Tehát csak a „Kérelmezők” csoport tagjai számára legyen érvényes az új GPO-unk. Ehhez a Management konzolon keressük meg az új GPO-unkat, jelöljük ki a struktúrában.



A GPO-hoz tartozó információk ablakban válasszuk a „Delegation” fület és a jobb alsó részénél található „Advanced” gombot nyomjuk meg. Itt vegyük fel a „Kérelmezők” csoportot, és állítsuk be úgy a jogosultságokat, hogy a GPO csak a „Kérelmezők” csoportra hajtódjon végre.



Miután ezzel megvolnánk, hangolhatjuk tovább a GPO-unkat. Érdekes néhány alapbeállítást megtennünk, hogy gördülékenyebben menjen a regisztráció a webform-on keresztül. Érdeemes letiltani a kilépcsőkor történő automatikus mentést, hogy mindig az alapsetet álljon elő a belépcsőkor.

Administrative Templates	
Policy	Setting
<b>Desktop</b>	
Don't save settings at exit	Enabled
<b>Desktop/Active Desktop</b>	
Active Desktop Wallpaper	Disabled
Disable Active Desktop	Enabled
Disallows HTML and Jpg Wallpaper	

Ugyanígy tiltani érdemes az „Active Desktop”-ot is. Persze ez nem a legkritikusabb beállítás, de izlés szerint végezhető, elvégre elvileg nem fog a felhasználó hozzáférni a „Desktop”-hoz amúgy sem.

Nem szabad megfeledkeznünk a belépési információk kezeléséről sem. A regisztráció alatt a felhasználónév és a hozzátartozó jelszó nyilvános, ezért ennek módosítási lehetőségeit is korlátozzuk. Illetve szabjunk némi gátat a nézelődésnek, és a futó taszkok közötti turkálásnak.

System/Ctrl+Alt+Del Options	
Policy	Setting
Remove Change Password	Enabled
Remove Lock Computer	Enabled
Remove Task Manager	Enabled

Már csak a „shell” csere maradt hátra. Ehhez engedélyezni kell a saját felhasználói „interface”-t a „system” cuccai között.

System	
Policy	Setting
Custom user interface	Enabled
Interface file name (for example, Explorer.exe)	%systemdrive%\program files\internet explorer\iexplore.exe -k https://info/accountmanagement



Bár elvileg minden kliens gép egyformán telepített, de lehetnek eltérő telepítések is a későbbiekben. Rájuk gondolva az IE elérési könyvtárra érdemes változó segítségével hivatkozni: „%systemdrive%\ Program Files \ Internet Explorer \ iexplorer.exe”. A „-k” kapcsoló biztosítja a teljes képernyős (FullScreen) megjelenítését az IE-nek. Legvégül pedig meg kell adnunk értéként az elindítani kívánt weblapot, ahol a regisztrációs form leledzik. Esetünkben ez a: <https://info/accountmanagement>.

Ha mindent jól csináltunk, és bejelentkezünk az egyik intézményi munkaállomáson az intézményhez tartozó kérelmező felhasználóval, már megérkezik számunkra a webform-unk teljes képernyőn a szokásos „Desktop” helyett. Kiöltöhetjük a regisztrációs lapot saját adatainkkal, és már megy is az automatikus levél az intézményi vezetőnknek hozzáférési kérelmünkkel jóváhagyásra. De hogyan is megy az a levél?

## Automatikus e-mail

Ez egyszerű. Ez a megoldás sem szentírás, ám bizonyos okok miatt egy adott kontextusban a legésszerűbbek közé tartozik. Tehát az előző számban elkezdett webform forráskódját egészítsük ki az alábbiakkal.

```
private void sendMail(UserManagement.UserData
uData)
{
    string[] mailAddresses;
    UserManagement.Credentials crds;
    ...
    Crds =
    (UserManagement.Credentials)Session["crds"];
    Cnfg =
    System.Configuration.ConfigurationSettings.
    AppSettings;
    mailAddresses =
    aUser.getOuMailAddresses(uData.Path);
    ...
    msg.Bcc =
    cnfg["mgmt.logmailbox"] + "; " +
    mailAddresses[0];
    msg.To = mailAddresses[1];
    ...
}
```

A web.config-ban már definiáltuk a levelező szerverünk helyét az előző számban. Ezért most ezt nem tesszük meg újra. A „...”-ok azért vannak kitéve a kódárszletben több helyen, mert ezeken a részekben lehet a hibákat kezelni, illetve a levél külalakját, egyéb dolgait rendezni, stb.

A „Cnfg” szerepe kiolvasni a web.config fájlból az ide vonatkozó bejegyzéseket. A „Crds” pedig az autentikáció eredményeként elmentett felhasználói adatokat szívja magába. A „mailAddresses”-nél egy tömböt kapunk eredményül, amibe az adatok az AD-ból érkeznek, tehát az „aUser” objektum getOuMailAddresses metódusa ad vissza egy „string” tömböt (megfelelő e-mail címekkel):

- A 0. elemben az archiválásra kijelölt felhasználónak konfigurált e-mail cím van elcsomagolva. Tehát ezen kijelölt személy megkap minden regisztrációs kérelmet, engedélyt, módosítást, stb. másolatban. Ez egy nagyon kezelhetőség érdekében érdemes szintén a

web.config fájlban definiálni. Jól látszik a kódárszletben, hogy a neki szánt üzenet BCC-ben megy.

- Az 1. elemben meg az adott intézményi OU vezetőjének (vagy engedélyezésre jogosultsággal rendelkező felhasználójának) e-mail címe van.

Magának az üzenet tartalmának a felépítése is igen egyszerűen definiálható ugyanebben a metódusban:

```
...
msg.Subject = "Új felvételi kérelem";
msg.Body = "Új belépési kérelem jött létre!" +
Environment.NewLine + "A kérelem a http://" +
Request.Url.Host + Request.ApplicationPath +
"/UserList.aspx oldalra hagyható jóvá, vagy
utasítható el.";
...
msg.Body += Environment.NewLine;
msg.Body += Environment.NewLine;
msg.Body += "Intézmény: ";
msg.Body += crds.InstituteName;
msg.Body += " (" + crds.InstituteOU + ")";
msg.Body += Environment.NewLine;
msg.Body += Environment.NewLine;
msg.Body += "Teljes név: ";
msg.Body += uData.FullName;
msg.Body += Environment.NewLine;
msg.Body += "Elsődleges e-mail: " +
uData.MailNickName;
msg.Body += Environment.NewLine;
msg.Body += "e-mail2: " + uData.Alias2;
msg.Body += Environment.NewLine;
msg.Body += "e-mail3: " + uData.Alias3;
msg.Body += Environment.NewLine;
msg.Body += "e-mail4: " + uData.Alias4;
...
```

A kódárszlet elején meghatározásra kerül az üzenet címe: „Új felvételi kérelem”. Ezt követően pedig az a http elérés rakódik össze darabokból, ahol a jóváhagyó engedéllyel rendelkező felhasználó pipát tehet a kérelmező neve mellé engedélye jeléül. Majd a következő sorokban, rendezett formában beíródnak az igénylő kitöltött adatai (neve, intézménye, igényelt e-mail címe, plusz három „alias” címe, stb.). Amennyiben jó a web.config fájlban a levelező szerver definiálása, akkor már mennek is az elektronikus levelek szakadatlanul engedélyezésre. A jóváhagyó felhasználónak meg nincs más dolga, mint a levélben lévő hivatkozásra kattintva pipát rakjon a kérelmező neve mellé, hogy a felhasználói fiók élesedjen az AD-ben.

## Felhasználói fiók engedélyezése

Az előző számban tárgyaltuk a felhasználói fiók létrehozásának mikéntjét webform-on keresztül. Csupán az az egy kérdés maradt tisztázatlan, hogy miként lesz „disable” az „account” a létrehozáskor, illetve hogyan élesedik, ha az engedélyt jelentő pipa rákerül a weblapra.

Amikor létrehozunk egy felhasználót, akkor egy rakás paraméterével együtt hozzuk létre (ezen paraméterek részünkre fontosabbjait átnéztük az előző számban). Ezek közé tartozik a „UserAccountControl” nevű tulajdonság is. Ez egy jó nagy bitmaszk, ami azt jelenti, hogy ez egy olyan szám, melynek

minden egyes bitje egy kapcsoló, és mindegyik egy pontosan meghatározható tulajdonságot kapcsolgat ki és be attól függően, hogy az a bit 1 vagy 0.

A felhasználó létrehozásakor ezt kell 514-re állítani. Ami a következőket jelenti:

- AccountDisable (2)
- normalAccount (512)

Érdekes a kódban (ahol az AD-vel kapcsolatos műveleteket végezzük) egyébként „enum” ként definiálni a UserAccountControl pár értékét a könnyebb kezelhetőség érdekében. Így mindig csak az eltéréseket kell megadnunk.

```
[Flags]
public enum UserAccountControl
{
    script = 1,
    accountDisable = 2,
    homedirRequired = 8,
    lockout = 16,
    passwdNotRequired = 32,
    normalAccount = 512,
    passwordExpired = 8388608
}
```

Tehát a továbbiakban csak egy AccountControl nevű tagváltozót kell szerepeltetnünk az egyszerűbb átláthatóság érdekében. Pl. ha a fenti leiltolt felhasználót létre szeretném hozni, akkor az alábbi sorokat kell használni:

```
public UserAccountControl AccountControl =
UserAccountControl.accountDisable |
UserAccountControl.normalAccount;
```

Ennek felhasználása egybeköthető az OU-ba való elhelyezéssel, mikor a felhasználó létrejön az igénylés alapján automatikusan (persze leiltolt állapotban).

```
public string SetUserRequested(string dn)
{
    DirectoryEntry userEntry;
    DirectoryEntry statusEntry;
    InstituteAndStatus IaS;
```

```
userEntry = new DirectoryEntry("LDAP://"+
+ dn);
IaS = getInstituteEntry(userEntry.Path);
if (IaS.Status != UserStatus.requested)
{
    statusEntry = new DirectoryEntry("LDAP://OU="
+ requestedOU + "," +
IaS.InstituteEntry.Properties
["distinguishedName"][0].ToString());
serEntry.MoveTo(statusEntry, userEntry.Name);
userEntry.Properties
["userAccountControl"][0] =
((Int32)userEntry.Properties
["userAccountControl"][0]) |
((Int32)UserAccountControl.accountDisable);
userEntry.CommitChanges();
}

return userEntry.Path;
}
```

A kódrészlet szintén magáért beszél. Jól látható benne, hogy az adott intézményi OU alá, a „Requested” OU-ba kerül a felhasználó, és a kódrészlet alkalmazza a „Disable” tulajdonságot. Innen már csak egy ugrás, hogy amikor a felhasználót engedélyezik, akkor szinte ugyanígy mozgassuk (a MoveTo segítségével) a felhasználónkat a „Ready” OU-ba. Persze töröljük egyúttal a „Disable” bitet a felhasználóhoz tartozó „mask”-ban.

Azt hiszem készen volnánk. Van bejelentkező webform-unk „shell” cserés megoldással, létrejön az igényelt felhasználói fiókunk, mennek az automatikus levelek, és végül, de nem utolsó sorban tudjuk a kérelmeket engedélyezni intézményenként külön-külön definiált jóváhagyók meghatározásával. © Persze a megoldás nem teljes, csupán sarokpontokat emeltem ki útmutatónak.

Aki ennél többet szeretne tudni a témáról, az bátran keresse fel az IQSOFT – John Bryce Oktatóközpont munkatársait.

FARKAS VIKTOR  
IQSOFT- John Bryce Oktatóközpont  
farkas\_v@btmail.com  
MCSE, MCT, HP-ASE

## Tanfolyami akciók!

Windows 2003 tanfolyamhoz **30% kedvezmény** az Exchange 2003 és SMS 2003 tanfolyamok árából!  
Kedvezményes MCSD fejlesztői tanfolyami csomagok.

### Új tanfolyamok!

SharePoint Portal Server 2003, Microsoft Operations Manager 2005, ISA 2004 Projektmenedzsereknek egynapos, technológiai áttekintést nyújtó előadások.

### Microsoft SA oktatási kuponok beválthatók

Nálunk beválthatja a Microsoft Software Assurance licenc vásárlása után kapott oktatási kuponjait!

**További információkért hívja munkatársainkat!**

**IQSOFT – John Bryce**  
OKTATÓKÖZPONT

**IQSOFT – JOHN BRYCE**  
OKTATÓKÖZPONT KFT.

Cím: 1135 Budapest  
Csata u. 8.  
Web: [www.iqb.hu](http://www.iqb.hu)  
Telefon: 236-6197-8  
E-mail: [tanfolyam@iqb.hu](mailto:tanfolyam@iqb.hu)

Microsoft  
Software Assurance  
**Microsoft**  
CERTIFIED  
Partner

Learning Solutions



# Dr. Watson

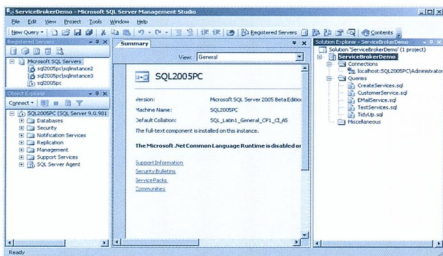
## Az SQL 2005 ÚJDONSÁGAI I. RÉSZ

Jön! Jön! Jön! Csak még azt nem tudjuk, mikor! De előbb-utóbb megújul az immár öt éves adatbáziskezelő, az SQL Server. E cikkben összeválogattam egy-két olyan újdon-  
ságot, ami messze túlmutat a kozmetikai változtatásokon. Ezek közül több akár egy teljes cikket megérdemelne, pedig a pusztá felsorolásuk is két cikkbe fér csak bele.

Kozmetikai újdon-ság is van persze: az `sp_addlogint` felváltja a `CREATE LOGIN`, az `sp_adduser` felváltja a `CREATE USER`, és általában minden parancs SQL-esebben fog kinézni. De lássuk a lényegét!

### A Solution Explorer

A felhasználói felületen kő követő nem maradt: odalett az Enterprise Manager, a Query Analyzer... Helyettük egy Visual Studio-szerű új kezelőprogramot kapunk, az SQL Management Studiot. Egyesek erre azt mondanák, hogy ez bizony kozmetikai változtatás a javából, de ki kell ábrándítanom őket: az SQL Management Studio nemcsak külsőleg hasonlít a Visual Studióra, hanem szolgáltatásaiban is. Nemcsak egyedi fájlokkal változtathatunk a javából, de solution-  
okat kezelhetünk. Aztán itt van a disconnected (lecsatolt) üzemmód: ha van a környéken SQL Server, ha nincs, lehet vele dolgozni. Így néz ki:



SQL Server Management Studio

Bal oldalon találjuk az Enterprise Manager maradékát, jobbszélén pedig a Solution Explorer, benne egy konnektóval, egy marék SQL-scripttel és nulla Misc-fájllal. Ez a Solution fontos szerepet fog kapni a következőkben. De először egy ügyes admin-dologról.

### DDL triggererek

Mostantól nemcsak DML (INSERT/UPDATE/DELETE) parancsokhoz lehet triggereket írni, hanem DML-ekhez is, (pl. CRE-

ATE TABLE). Ezáltal az ilyen műveleteket naplózni, illetve visszavonni (ROLLBACK) is képesek leszünk.

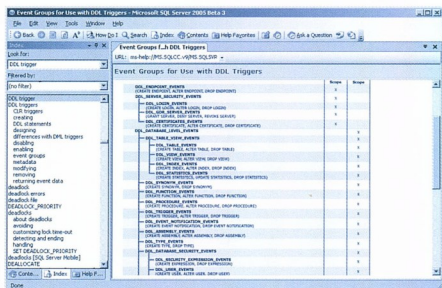
```
CREATE TRIGGER Nemszabad
ON DATABASE
FOR DROP_TABLE, ALTER_TABLE
AS
PRINT 'Sajnos táblát nem szabad törölni vagy
módosítani!'
ROLLBACK
GO
```

Lehet próbálkozni a törléssel:

```
DROP TABLE Fontos_Adatok
```

Nem fog menni!

A fenti trigger mutatja az alapokat, ugyanakkor ez a lehetőség még rendelkezik tartalmakkal. Idepasztázom a Books Online ábráját a hierarchikusan egymásra épülő DDL-triggererekről:



### DDL trigger fajták

Hát nem egyfélé van... Menjünk is tovább, ne tátsuk a szánkát.

## Service Broker-architektúra

Régi vágya teljesül a programozóknak ezzel az alapvetően aszinkron műveletvégző rendszerrel. Még a múlt évezredben szükségem lett volna egy olyan szolgáltatásra, amivel a kliensprogramokat ki tudtam volna értesíteni egy táblaérték változásáról (tözsdei árfolyamok változása). Akkor sem, és az azóta elmúlt tíz évben sem volt erre a feladatra kultúrmelegedés. Triggerrel nem lehet megoldani, mert akkor a munkaállományok kiértékelése a tranzakció részévé vált volna. Ennél nagyobb butaságot nehéz lenne elkövetni. Hát akkor? Pollingoljunk? De milyen gyakran? Szinte mindegy, a megoldás úgyis rossz lesz. Ha gyakran kérdezi le a sok-sok munkaállomány ugyanazt az SQL Servert (ahol ráadásul nem is fel-tétlenül történt változás), a hálózati sávszélesség látja kárát. Ha ritkábban, akkor meg a felhasználók háborognak, hogy nem elégé friss az adat.

Kellene ide valami aszinkron-trigger!

Ezt a feladatot csodálatosan meg lehet oldani a Service Broker-architektúra felhasználásával, ami nem más, mint egy várakozási sor alapú, tranzakcionál, aszinkron végrehajtási rendszer az SQL Serveren belül. Tetszőleges, akár kritikus fontosságú műveletek (trigger, tárolt eljárás) indíthatnak akár órákig tartó folyamatokat, a rendszer mégsem áll be, mert csupán egy parancs kerül egy várakozási sorba, amit azután az SQL Server végrehajt.

Az egész Service Broker-dolgot TSQL-ben lehet megvalósítani. Csupán néhány utasítás kell egy ilyen „gép” elkészítéséhez (CREATE QUEUE, CREATE MESSAGE TYPE, CREATE SERVICE stb.) Lássunk ezekre néhány konkrét példát!

```
CREATE MESSAGE TYPE Uzenet
    VALIDATION = WELL_FORMED_XML
CREATE QUEUE Sorocska
CREATE SERVICE Kiszolgalo ON QUEUE Sorocska
```

És így tovább. Úgy kell ezt elképzelni, mint a szerveroldali kurzorokat. 60-80 TSQL-utasítás, és már zakatol is!

Akaszják a hóhért: avagy a Microsoft mire használja ezt az új arendszerét? Lásd alább!

## SQL Notifications

Az SQL 2005-ben van egy szolgáltatás, ami lehetővé teszi, hogy különböző SQL-események (DDL, DML, DCL) hatására automatikusan meghívódjon egy VALAMI, és feldolgozza az eseményt. Ennek neve SQL Notifications, és teljes egészében a fentebb tárgyalt Service Brokerrel hozták létre. (Ez nem keverendő össze a Notification Servicessel, ami kifejezetten arra való, hogy jüzeket értesítsünk emailben/faxon arról, hogy új termék érkezett.) A DDL triggerekkel sem keverendő össze, mert azok BENNE vannak nyakig a tranzakcióban, tehát akár vissza is vonhatják (ROLLBACK), viszont a világ megváltására, a globális felmelegedés megállítására nem lehet felhasználni őket, mert ez mind a tranzakció részét képezné. Az SQL Notification-üzenet feldolgozása nem része a tranzakciónak, így akár kiszámíthatja a 234223424324 faktoriálisát is, mégsem akasztja meg a tranzakciókat. Ennek az előnynek az ára az, hogy nem tud beavatkozni a hívó folya-

matba pl. egy ROLLBACK utasítással. A végrehajtandó VALAMI tényleg tetszőleges program lehet, tehát külső is, sőt, távoli is. A legegyszerűbb persze a TSQL "fogyasztó", erre mutatok be egy példát:

```
-- enable Service Broker
USE MyProductCatalog
ALTER DATABASE MyProductCatalog SET ENABLE_BROKER;
```

A bróker adatbázisonként nyilván csak egyszer kell engedélyezni. Most jön a lényeg: négy új CREATE utasítás! (Lokális TSQL-fogyasztó esetén nem látjuk elsőre, hogy minek ez a hókuszpókus, de távoli EXE fogyasztónál már belátnánk, hogy kell „kjú”, kell „rút” stb.)

```
-- create a queue
CREATE QUEUE NotifyQueue
GO

CREATE SERVICE NotifyService
ON QUEUE NotifyQueue
(
    [http://schemas.microsoft.com/SQL/Notifications/
    PostEventNotification]
)
GO

CREATE ROUTE NotifyRoute
WITH SERVICE_NAME = 'NotifyService',
ADDRESS = 'LOCAL'
GO

CREATE EVENT NOTIFICATION NotifyCREATE_TABLE
ON DATABASE
FOR CREATE_TABLE, ALTER_TABLE
TO SERVICE [NotifyService]
```

A fentiek egy „kjúba” fogják pakolni az összes CREATE TABLE és ALTER TABLE műveletet. És itt ismét látjuk, mi a különbség a Service Broker és a DDL Triggerek között: ez csak kiértékel, de a tranzakció már megtörtént.

Kell még egy "fogyasztó", ami a jelen esetben csak kiírja az esemény XML-jét:

```
DECLARE @messageTypeName NVARCHAR(256),
        @messageBody XML
RECEIVE TOP(1) @messageTypeName =
message_type_name, @messageBody = message_body
FROM dbo.NotifyQueue;

IF @@ROWCOUNT = 0
RETURN

PRINT CONVERT(NVARCHAR(1000), @messagebody)

DECLARE @cmd NVARCHAR(1000)
SET @cmd =
CONVERT(NVARCHAR(100), @messagebody.query('data(//T
SQLCommand//CommandText)'))
```

Az SQL mailt felváltó, SMTP-alapú SQLiMailt is Service Brokerként valósították meg.



## Dynamic Management Views

Az eddig is létező syslocks "memória-táblán" feül további nézetek vannak az SQL 2005-ben dinamikusan változó, tehát táblában NEM tárolt adatok lekérdezésére. Mit nem tárol táblában? Hát például a zárolásokat, mert ettől mintegy 100-szor lassabb lenne. Meg a végrehajtási szálak (thread) információit sem.

A Dynamic Management nézetek nevében az egységesség jegyében ott találjuk a dm\_ rövidítést. Innen tudhatjuk, hogy mikor turkálunk csupán a memóriában. Ime néhány:

```
sys.dm_db_partition_stats
sys.dm_exec_sessions
sys.dm_io_pending_io_requests
sys.dm_os_memory_pools
sys.dm_os_threads
sys.dm_broker_queue_monitors
sys.dm_tran_locks
```

Az igazsághoz hozzátartozik, hogy néhány sys.dm\_ kezdetű csoda nem view, nem is tárolt eljárás, hanem függvény, mégpedig azért, mert különböző paramétereket vár a futtatáshoz.

## Vendégoszlopok az indexben

Annál nagyobb a lekérdezések teljesítménye, minél kevesebb adatot kérdezzünk le, ez eddig nyilvánvaló. Ha annyira kevés adatot kérdezzünk, hogy csak az indexkulcsokat, akkor le sem kell menni az adatokig, és a lekérdezés eredménye már elő is állítható az indexből magából. Ezt hívják fedő indexnek (covered query), amelyeknek a sebességen túl még az az előnye, hogy mivel az adattáblát nem kell feltérni, semmilyen tábla, lap vagy sorzárolást nem okoz.

Az SQL 2005 újdonsága, hogy a fedő nagyobb lehet, ugyanis az indexelt mezőkön kívül további mezőket is lehet emelni az indexbe, "csak úgy", mindenféle sorbarendezés kényszer nélkül. Így az index akkor is fed, amikor nem fed.

## XML adattípus, XML indexek

Elgondolkodtam azon, minek ez az XML-adattípus, XML-mező nekünk. Rájöttem: teljesítményokokból fogjuk használni, mi, mindannyian! Vegyünk egy példát: van kétszáz ezer rekordunk, mindegyikhez tartozik X darab jellemző, amelyeket csak a normalizálás igénye miatt teszünk külön táblába, mert hogyan is tárolnánk másképp egy one-to-many viszonylatot? A Jellemzők tábla jelenleg 1 millió 600 ezer soros.

De most gondoljuk végig, hogyan lehet egy bizonyos aparcordhoz összeszedni a gyermekeit? Hát dzsoinnal! Tételezzük fel, hogy van egy jó kis nonclustered index az APAID-re, ekkor apánként mindössze 10-12 darab 8k-s lap megfogásával hozzá is jutunk a jellemzőkhöz. (Aki nem látja világosan, miért lesz több mint 10 page IO, az jöjjön SQL tuning tanfolyamra...)

De hát ez borzasztó! Borzasztóan teljesítményigényes! És ha több száz, vagy ezer apához kérdeznénk le egyszerre? Eljönne az a pont, amikortól az SQL Server lemond az indexhasználatról, mert nemhogy gyorsítana a lekérdezésre, hanem egyenesen lassít!

Mit nyújt ezzel szemben az XML mezőtípus? Lehetővé teszi, hogy az apta tulajdonságai ott legyenek HELYBEN, bent a rekordban! Hány page IO-többletet jelent, ha a jellemzőket is elő akarom szedni? NULLÁTI! Ott van a kezemben!

Nyilván hátránya is van a dolognak, az XML-mezőben meg-

búvó "gyermekrekordokkal" dzsoinnolni más táblához, hát, hogy is mondjam: kihívás. Valamit valamiért!

Ez a gondolatmenet viszont egy dolgot sejtet: már megint tanulni kell, mégpedig sokat.

Csak példaképpen néhány szabvány, amit be kell magolni: XML séma, XPath, XQuery

Maga az XML adattípus szintén tartalmaz meglepetéseket: metódusai vannak! (Különböen hogy kitorászánk benne?) QUERY, VALUE, EXIST és MODIFY metódusai vannak a nyomorultnak, mindegyik mögött egy külön világ lakozik. Elírmisszerűen egy .QUERY, ahol xmCOL egy XML típusú mező:

```
SELECT xmlCol.query('declare default namespace =
"http://schemas.adventure-
works.com/InvoiceList";
<InvoiceNumbers>
{
  for $i in /InvoiceList/Invoice
  return <InvoiceNo>
    (number($i/@InvoiceNo))
  }
')
</InvoiceNumbers>')
```

Az XML indexek világa is érdekes, és teljesen különbözik a hagyományos indexektől. Itt többféle indexet lehet egymásra építeni. Az alap XML index nem különbözteti meg egymástól az útvonalakat, értékeket és attribútumokat, noha amikor XQueryyzünk, ezekre külön-külön hivatkozunk. Úgyhogy az alap XML indexre további háromféle index ültethető: PATH, PROPERTY és VALUE. Az alap (elsődleges) XML index létrehozása:

```
CREATE PRIMARY XML INDEX idxItem
ON Sales.Invoices(ItemList)
```

Erre "rűletünk" mondjuk egy PROPERTY indexet:

```
CREATE XML INDEX idxItemProp
ON Sales.Invoices(ItemList)
USING XML INDEX idxItem
FOR PROPERTY
```

És kész!

## Nyelvegységésítés, Schema

Ez a változtatás elsöre kozmetikainak tünik, noha a jogosultságkezelésben alapvető változtatásokat hoz: immár az SQL Serverben is beszélhetünk öröklődő jogosultságokról! A schema szintjén kiadott jogosultságok öröklődnek a benne lévő objektumokra.

Átalakul az objektumok elérési útvonalai is: a teljes névben a középső tag már nem a felhasználó neve, hanem egy úgynevezett schema, ami egy "konténer" az adatbázison belül, ebbe „döbálhatjuk” az együtt kezelendő objektumokat.

```
Select a,b,c from server.database.schema.table
```

Ezzel még lesz probléma, mert a régi scriptekben természetesen a felhasználó szerepelt az útvonal közepén. Kompatibilitási okokból az alapértelmezett schema neve: dbo...

## Adatok titkosított tárolása, tanúsítványkezelése

Tanúsítványkezelés TSQL-ből:

```
CREATE CERTIFICATE AWCert
WITH SUBJECT = 'SalesCert'
, ENCRYPTION_PASSWORD = 'Pëssw0rd'
```

A fenti utasítás létrehoz egy önálírt x.509 tanúsítványt, amit titkosításra lehet használni. Azt egyelőre nem tudom, hogy hogyan, de lesz ilyen az SQL 2005-ben. Amint felderítettem a terpet, jelentkezem a témával.

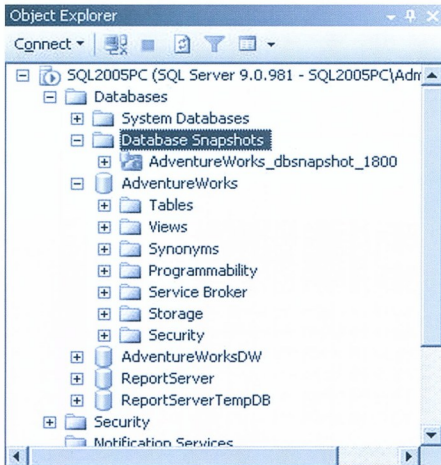
## Database Snapshot

Szinte minden tárolószolgáltatás képes már pillanatfelvétel készítésére. Mostantól az SQL Server is képes lesz erre, aminek nagy előnye, hogy bármikor hozzájuthatunk egy adott pillanatban konzisztens adatbázis tartalomhoz. Maga a pillanatfelvétel egy külön adatbázisba kerül. Ha a korábbi állapotra vagyunk kíváncsiak, ebben kell futtatni a lekérdezéseinket.

A Snapshot folyamatos üzemben készül az éles adatbázisról: mielőtt egy 8k-s lap változik, az érintetlen eredeti lap átmásolódik egy másik adatbázisba. Ehhez egy speci CREATE DATABASE utasításra lesz szükségünk:

```
CREATE DATABASE AdventureWorks_dbsnapshot_1800
ON (NAME = AdventureWorks_Data,
FILENAME = 'C:\AdventureWorks_Data.mdf')
AS SNAPSHOT OF AdventureWorks
```

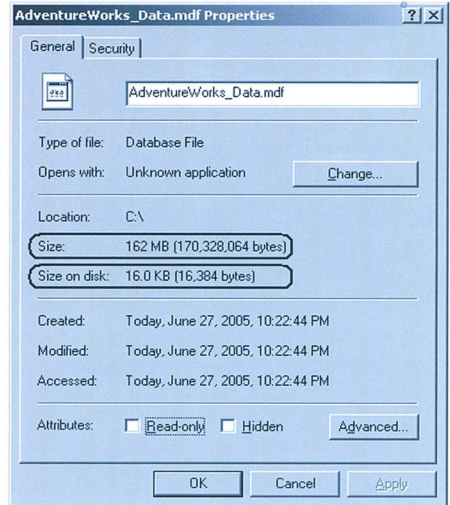
A lényeg a végén van: "AS SNAPSHOT OF egy\_másik\_db"



## Snapshot adatbázis a Management Studioban

Mekkora lesz ez a "mentés"? Látszólag (DIR \*.mdf) ugyanakkora, mint az eredeti, noha - amíg változás nem történik - kéremény 16 kilobájt van benne. Tényleg. Mivel ez egy NTFS

Sparse File. (Ez az első szolgáltatás, amiről tudok, hogy kihasználja a Sparse Filest)



## Sparse File látszólagos és fizikai mérete

Persze ha valaki egy hexaaditonnal megnyitja ezt a fájlt, akkor mindig a 162 megabájtot "végig tudja olvasni", úgy jönnek belőle a nullák, mint a sicc!

És ebbe bele lehet szelektálni már most, hogy még üres! Az alábbi két SELECT most még azonos eredményhalmazt ad. (Hogy lehet ez? A snapshot nem üres? De az, viszont az SQL Server átirányít minket az eredeti adatbázisra, ha "hiányzó" 8k-s lapot kérdezzünk le.) Persze ha módosítjuk az eredetit, az eltérés nyilvánvaló lesz.

```
USE AdventureWorks
SELECT *
FROM HumanResources.Employee
GO
USE AdventureWorks_dbsnapshot_1800
SELECT *
FROM HumanResources.Employee
GO
```

Ha valaki véletlenül töröl, módosít stb. az eredetiben, egy sima SELECT-INSERT vagy UPDATE segítségével visszarántja az eredeti adatokat a snapshotból.

Aprópó: mi az az AdventureWorks? Az új példaadatbázis. Northwind és Pubs has gone. Folytatása következik...

FÓTI MARCELL  
MCT, MCSE, MCSA  
marcellf@netacademia.net



## A NETACADEMIA KFT. a 2005-ös tanévre a következő *tanfolyamokat* ajánlja:

### RENDSZERGAZDAI TANFOLYAMOK

#### 2433 + 2439\*

VB Scripting, Windows Script Host Essentials  
+ Using WM

#### 2821\*

Designing and Managing a Windows Public  
Key Infrastructur

#### 2810 + 2830\*

Designing Security for Microsoft Networks

Oktató: **Fóti Marcell**

(MCSE, MCSA, MCT, MCDBA – 1995 óta)



### FEJLESZTŐI TANFOLYAMOK

#### DP\*

Objektumorientált tervezés Design Patternekkel .

#### 2030

Riportok készítése az SQL Server 2000 Reporting Services-szel

#### 2734

Updating Your Database Development Skills to Microsoft SQL Server 2005

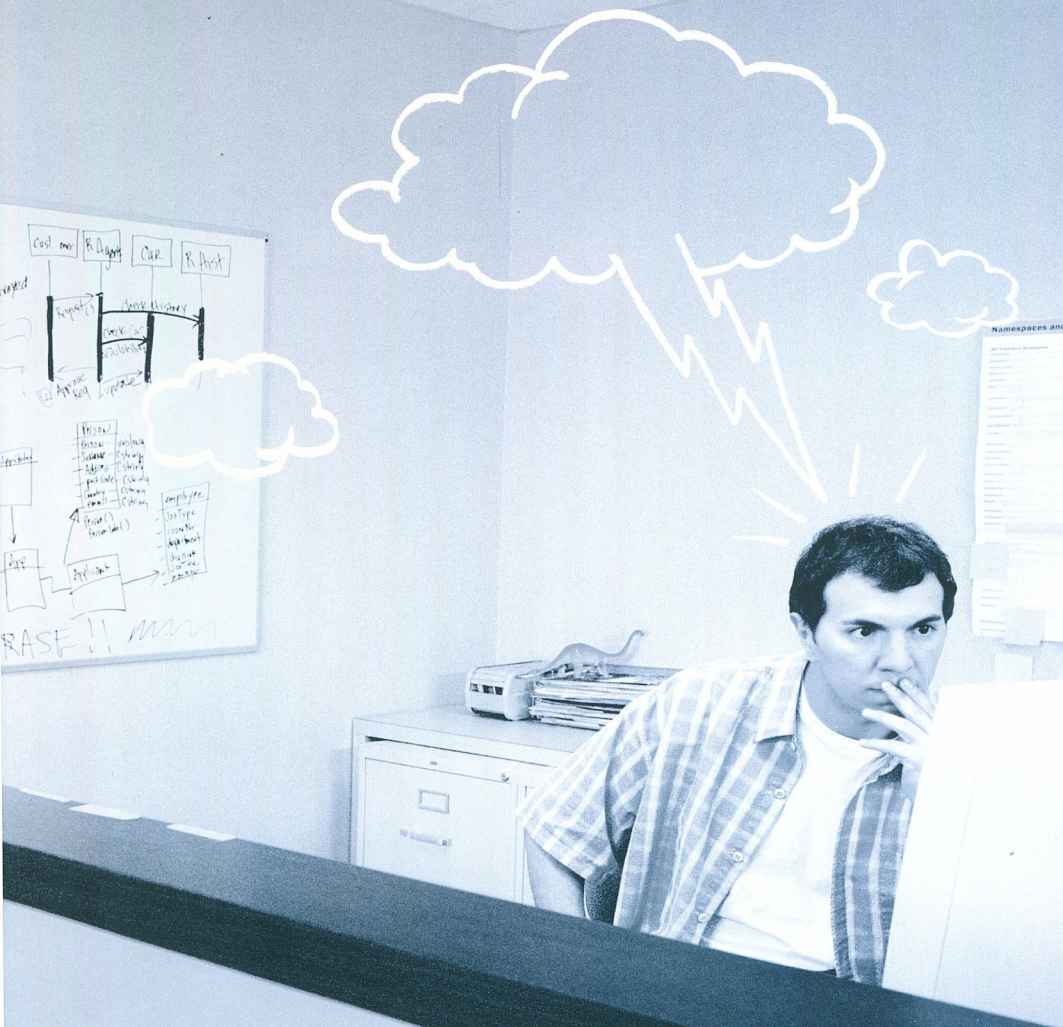
Oktató: **Soczó Zsolt** (MCSE, MCSD, MCDBA, MCAD, MCT, MVP)

Jelentkezni 06 1 472-1215-faxszámon vagy az on-line jelentkezési lap kitöltésével lehet.  
A letölthető és az on-line jelentkezési lapot a <http://www.netacademia.net> címen találja.

NetAcademia Oktatóközpont

1062 Budapest, Andrásy út 62. • Telefon: 06 1 472-1214 • Fax: 06 1 472-1215

\* A tanfolyam a szakképzési hozzájárulás terhére elszámolható (A NetAcademia Kft. nyilvántartási száma a Fővárosi Munkaügyi Központ által kiadott értesítés szerint: 01-0707-04.).



## MI MÁR LÁTJUK,

ahogy a következő **NAGY ÖTLET** megszületik.

Egy fejlesztőnek az ötlet már önmagában siker. Épp ilyen fontos, hogy ezek az ötletek a mindennapi életben is megvalósuljanak. Ezért teszünk meg mindent, hogy a fejlesztők kezébe olyan szoftvereket adjunk, amelyekkel megvalósíthatják elképzeléseiket. Az ötleteket, amelyekkel később mindenki nyer.

*Neked lehetőség. Nekünk kihívás.*

Your potential. Our passion.™

**Microsoft**