# TechNet MAGAZIN

2007. ÁPRILIS – MÁJUS

MÉLYVÍZ, CSAK ÚSZÓKNAK!

Windows Server® 2008

## ERŐS BÁSTYA
Számtalan biztonsági újdonság érkezik az új Windows Serverben

## OLAP-ALAPOK
Összetett kimutatáskészítő rendszer építése

## SAJÁT MICROSOFT
Interjú Kőnig Tiborral

## MILYEN KÉPESSÉGEK ÉS ESZKÖZÖK ÁLLNAK RENDELKEZÉSÜNKRE AZ ÚJ RENDSZERBEN?

## AD – CSAK FRAPPÁNSAN

## MAXIMUM ÉS MINIMUM – A SERVER CORE

## WINDOWS SERVER VIRTUALIZATION

## ÖSSZEFŰZVE – ÚJ HÁLÓZATI TECHNIKÁK

Microsoft TechNet

# Minden 7. nyer!

## Legyen Ön a HETEDIK és visszanyerheti tanfolyam díjának 50%-át!

**Most érdemes részt venni hivatalos Microsoft tanfolyamainkon, mert nyári akciónk keretében tanfolyamonként minden HETEDIK jelentkezőnek a képzési díj 50%-át visszaadjuk szabadon felhasználható tanfolyami utalvány formájában.**

**Néhány képzésünk ízelítőül június-július hónapra:**

- Windows 2003 Active Directory üzemeltetés (2279) – *június 4-8.*
- Windows XP üzemeltetési ismeretek (2272) – *június 4-8.*
- SQL 2005 adatbázis-adminisztráció, üzemeltetés (2780+2789) – *június 11-14.*
- Exchange Server 2003 üzemeltetés (2400) – *június 11-15.*
- Windows 2003 hálózat rendszertervezés, optimalizálás (2278) – *június 11-15.*
- Windows Vista üzemeltetés (5115-5117) – *június 11-15.*
- Windows 2003 hálózat üzemeltetés (2276/2277) – *június 18-22.*
- Windows Server 2003 üzemeltetési ismeretek (2273) – *június 25-29.*
- Windows XP terméktámogató, help-desk képzés (2261+2262) – *július 2-6.*
- SQL 2000/2005 alapok, lekérdezések (2071/2778) – *július 2-3.*
- SQL 2005 adatbázis implementáció, programozás (2779) – *július 4-6.*
- Microsoft Operation Manager 2005 üzemeltetés (2287) – *július 2-4.*
- Windows 2003 Active Directory üzemeltetés (2279) – *július 9-13.*
- SQL 2005 adatbázis-adminisztráció, üzemeltetés (2780+2789) – *július 30-augusztus 2.*
- Windows Server 2003 üzemeltetési ismeretek (2273) – *június 25-29.*

**Kísérje figyelemmel további induló képzéseinket (Tanfolyamlesünket) is weboldalunkon! Regisztráljon online, vagy juttassa el letölthető jelentkezési lapunkat munkatársainknak! Több tanfolyami jelentkezés, nagyobb esély!**

a mi tudásunk az Ön sikere

*Az akció valamennyi hivatalos Microsoft tanfolyamra érvényes, a tanfolyamok indulása esetén. A tanfolyamokra általános tanfolyami feltételeink érvényesek. Az akció más akciókkal, kedvezményekkel, kedvezményes konstrukciókkal nem vonható össze, az oktatási utalványok más szolgáltatásra vagy készpénzre nem válthatók át. Az utalványok felhasználási ideje 2007. december 20. Az akció augusztus 6-ig érvényes. További részleteket internet oldalunkon találhat. A képzésekre a jogszabályok szerint igénybe vehető a szakképzési hozzájárulás és felhasználható az SA oktatási utalvány.

Telefon: 203-0304/4122 m.

www.szamalk.hu/tisza

SZÁMALK

# ÉVÉRTÉKELÉS

## A megújult TechNet Magazin első éve.

**Budai Péter**
*Microsoft Magyarország*

Az új lapszámokat részben korábbi visszajelzések, részben saját elképzeléseink alakították olyanra, amilyen végül lett. Ideje volt azonban megkérdeznünk, hogy ez mennyire felel meg valójában az olvasók igényeinek. A legutóbbi számunkban található értékelőlapra számtalan visszajelzés érkezett, sokan arra is vették a fáradságot, hogy saját gondolataikat is megosszák velünk a számszerű értékelés mellett. Mi pedig – bár néha nehezen hiszik ezt el nekünk – sok időt szántunk arra, hogy a visszajelzések alapján kitaláljuk, merre tovább, illetve megvizsgáljuk, mennyire teljesítettük a kitűzött feladatot: hogy a leghasznosabb informatikai magazin legyünk a Microsoft-technológiákkal foglalkozó szakemberek számára.

Nézzük tehát, milyennek látják az olvasók a TechNet Magazint:

| | |
|---|---|
| *A TechNet Magazin összesített értékelése:* | *7,51/9* |
| *A tartalom minősége, stílusa, kidolgozottsága:* | *7,67/9* |
| *Hasznosság a szakmai tudás frissítésében:* | *6,93/9* |
| *Arculat, design, kinézet:* | *8,25/9* |
| *Ajánlaná ismerősének a Magazint?* | *A válaszadók 98,7 százaléka* |

A jelek szerint tehát összességében megfelelő úton járunk. Természetesen nem gondolom azt, hogy ezeken az értékeléseken ne lehetne javítani a jövőben, különös tekintettel a tartalom hasznosságára. Ezzel kapcsolatban volt még egy kérdésünk, mégpedig az, hogy mennyire elégedettek olvasóink a TechNet Magazin tartalmának mélységével: több áttekintés vagy több mélyebb cikk volna fontosabb? Itt az 5-ös érték jelentette volna azt, hogy most minden úgy jó, ahogy van. Az eredményül kapott 5,84 azt mutatja, hogy kicsivel több mély cikkre van igény, de nagyjából megfelelő az egyensúly.

Ezt az évet leginkább az új termékek áttekintésére szántuk. Hogyan lesz a folytatás? Ősztől szeretnénk az elmúlt évben bemutatott számtalan új Microsoft-szoftvert – soha korábban nem jelent meg ennyi termékünk egyszerre! – mélyebben is feltárni, hogy még többet segítsünk az informatikai szakemberek mindennapi munkájában. Sokkal több konkrét üzemeltetési területtel, és lényegesen több azonnal is használható tippel és trükkel jelentkezünk majd, de a jövőben is szeretnénk megfelelő áttekintést adni az új és érkező megoldásokról. Ezeket a cikkeket pedig továbbra is az ország legjobb szakemberei írják majd!

Most még ebben a számban egy kicsit előreszaladunk, és megmutatjuk, mi várható a Windows Server 2008-ban (ami hosszú-hosszú ideig Longhorn Server néven létezett a szakmai köztudatban). Ősztől ezeket és a korábban bemutatott technológiákat fogjuk tovább boncolgatni, reméljük minden olvasónk legnagyobb megelégedésére. Végezetül szeretnénk mindenkinek megköszönni a visszajelzéseket, és azt, hogy hűséges olvasóink!

TechNet MAGAZIN

**SZERKESZTŐSÉG**
**Főszerkesztő**
Sziebig Andrea – *asziebig@vogelburda.hu*
**Szakmai lektor**
Budai Péter – *i-pbudai@microsoft.com*
**Vezető szerkesztő**
Varga János – *jvarga@vogelburda.hu*
**Nyomdai előkészítés**
Budakeszi Bejárat Kft.
**Korrektor**
Matula Zsolt
**Lapterv és címlap**
Emotion Bt.

**Szerkesztőség és kiadó címe:**
Vogel Burda Communications Kft.
1077 Budapest, Kéthly Anna tér 1.
Tel.: 888-3400, fax: 888-3499

**KIADÓ**
**A Microsoft Magyarország megbízásából kiadja a Vogel Burda Communications Kft.**

**A kiadásért felel**
Walitschek Csilla
*cswalitschek@vogelburda.hu*
Tel.: 888-3450, fax: 888-3499



**Hirdetési igazgató:**
Farkas Viola – *vfarkas@vogelburda.hu*, tel.: 888-3459
**Médiareferensek:**
Németh Krisztina – *knemeth@vogelburda.hu*, tel.: 888-3468
Oláh Bernadette – *bolah@vogelburda.hu*, tel.: 888-3475
Rátóti Sarolta – *sratoti@vogelburda.hu*, tel.: 888-3453
Szendrey Szilvia – *szendrey@vogelburda.hu*, tel.: 888-3455
Fax: 888-3459
**Hirdetési koordinátor:**
Szőke Erika – *eszoke@vogelburda.hu*
Tel.: 888-3411, fax: 888-3459
**Nemzetközi hirdetésfelvétel:**
Eric N. Wicha – *ewicha@vogelburda.com*
Vogel Burda Holding
Poccistrasse 11, D–80336 München
Tel.: +49 89 74642-326, fax: +49 89 74642-325
A hirdetések körültekintő gondozását kötelességünknek érezzük, de tartalmukért felelősséget nem vállalunk.
**Marketing:**
Gajdos Barna – *bgajdos@vogelburda.hu*, tel.: 888-3494

**Terjesztés**
**Terjesztett példányszám:** 3000

**Nyomda:**
Pauker Nyomdaipari Kft.
1047 Budapest, Baross utca 11-15.
Felelős vezető: Vértes Gábor ügyvezető igazgató
**ISSN 1586-5185**

# Címlapon

## Biztonság



## Alkalmazásplatform



## Közösség

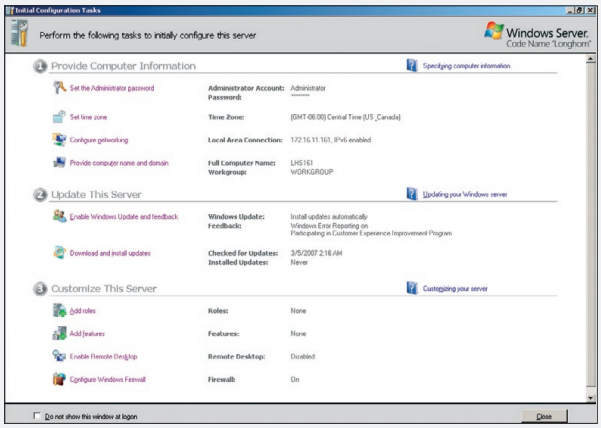
# Az első lépések

## Hogyan kezdjünk hozzá ennek az igencsak terjedelmes és nagy tudású rendszernek a megismeréséhez?

A legjobb megoldás az olvasáson kívül, ha megnézzük, milyen komponenseket telepíthetünk fel, vagyis, hogy egyáltalán milyen képességek és eszközök állnak rendelkezésünkre az új szerver-operációsrendszerben.

A telepítés során túl sok meglepetés még nem ér minket: funkcionalitás szempontjából ugyanis csak azt választhatjuk ki a grafikus felületen, hogy milyen verziójú Windows Server 2008-at telepítünk. Itt kell eldöntenünk, hogy Core vagy teljes Windows Server 2008-at szeretnénk, valamint a rendelkezésünkre álló szériaszám alapján itt automatikusan is eldőlhet, hogy azon belül Standard, Enterprise, Datacenter vagy Web Editiont kapunk. (Érdekesség, hogy a Web Edition most már sokkal hangsúlyosabb nevet kapott: ezentúl Windows Web Server névre hallgat majd ez a Windows Server 2008-változat.)

Mint már megszokhattuk, ha nem használunk a telepítéshez valamiféle válaszfájlt (unattend.xml), a szerver csak a legfontosabb, a működéséhez és elindulásához feltétlenül szükséges komponensekkel települ, és gyakorlatilag semmilyen szerepkör és képesség nem aktív ilyenkor még rajta. Azonban a rendszer első indításakor rögtön az itt látható képenyő fogad minket.




*Ez az ablak nem más, mint az új Server Manager egy nézete (Initial Configuration Tasks)*

Maga a Server Manager a korábbi Manage Your Server funkció helyett került be a rendszerbe, és gyakorlatilag nem más, mint a szerverünk kényelmes konfigurálásához használható MMC snap-inek gyűjteménye. Ez is, csakúgy, mint a Manage Your Server, alapbeállítás szerint a szerverre történő belépéskor automatikusan elindul (persze kikapcsolhatjuk, akár csoportházirendből is), azonban annyi különbség mindenképpen van, hogy ez már valóban hasznos adatokat, funkciókat tesz elérhetővé – ezért vélhetően kevesebben fogják reflexszerűen kikapcsolni.

Természetesen szeretnénk telepíteni a még szinte üres szerverünkre különféle képességeket. Itt érhet minket az első meglepetés, ugyanis szerepkörök (Roles) és képességek (Features) hozzáadására is van lehetőség. Vajon mi köztük a különbség? Miért van ezekre szükség?

Már a Windows Server 2003 esetén is volt mód szerepkörök telepítésére, azonban az még csak egy egyszerű varázsló volt arra, hogy a számunkra fontos, összetartozó képességek csoportját egyszerre, egy lépésben telepítsük a rendszerre, vagy ugyanezeket egy lépcsőben távolítsuk el. Windows Server 2003-nál azt is megtehettük, hogy ugyanezt az Add/Remove programs menüpontból, manuálisan érjük el. A következő meglepetés akkor érhet minket, amikor az Add/Remove programs opciót keressük: helyette ugyanis szintén a Server Manager köszön vissza ránk! Vagyis kénytelenek leszünk megbarátkozni – az amúgy nagyon logikusan felépített – szerepkörök és képességek listájával, nincs már más mód funkciók és komponensek telepítésére.

### A szerepkörök

A szerepkör (Role) egy olyan jól definiált feladategyüttes, amit szeretnénk, ha a szerverünk ellátna a hálózatunkon. A legtöbb szerepkörhöz azonban további szerepkör-szolgáltatások (Role Services) tartoznak, és ezek nem feltétlenül szükségesek az adott szerepkör működéséhez, hanem teljességgel opcionálisak, további képességeket tesznek elérhetővé. Valamennyi szerepkör-szolgáltatás tételes felsorolása nagyon hosszú volna, de azt érdemes áttekintenünk, hogy milyen szerepkörök állnak rendelkezésünkre:

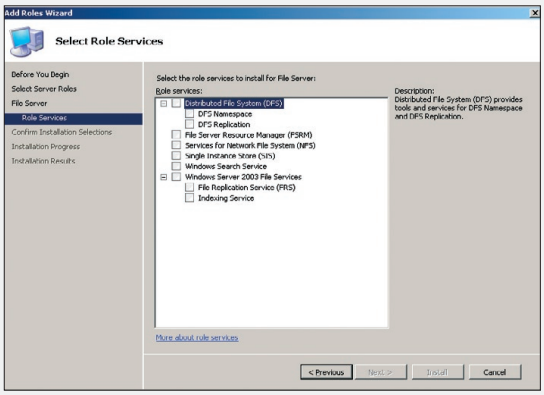
- Active Directory Domain Services;
- Active Directory Lightweight Directory Services;
- Active Directory Federation Services;
- Active Directory Rights Management Services;
- Active Directory Certificate Services;
- File Server;
- Print Services;
- Fax Server;
- DNS Server;
- DHCP Server;
- Network Access Services;
- Web Server (IIS);
- Windows SharePoint Services;
- Universal Description, Discovery, and Integration (UDDI) Services;

- Windows Deployment Services;
- Terminal Services;
- Windows Media Services.

A szerepkörök többsége néhány kivételtől eltekintve már ismerős lehet. Az Active Directory Lightweight Directory Services (AD/LDS) az Active Directory Application Mode (AD/AM) új neve. A Network Access Services többek között a Network Access Protection technológiát rejti magában, a Windows Deployment Services pedig az Automated Deployment Services (ADS) leszármazottja.

Nézzünk egy példát a szerepkör-szolgáltatásokra is! Példaképpen kiválasztjuk a File Server szerepkört, és megjelennek azok a további szolgáltatások, amelyeket ennek részeként telepíthetünk rendszerünkre.

Valamennyi itt szereplő szolgáltatás már teljesen opcionális, egyiket sem szükséges telepítenünk ahhoz, hogy az alap fájlszerver-funkciókat elérjük.




*A File Server szerepkör további szerepkör-szolgáltatásai*

Esetünkben telepíthetnénk például a File Server Resource Managert, amellyel hozzáférhetnénk ennek a konzoljához (fsrm.msc), és beállíthatnánk a felhasználóknak tárhely-kvótákat, értesítéseket, figyelmeztetéseket, fájltípus-megkötéseket, illetve generálhatnánk jelentéseket is. Ugyancsak kérhetnénk a Distributed File System szolgáltatás telepítését, ami viszont további két alszolgáltatásra bomlik, a DFS-N-re, illetve a DFS-R-re.

Ha kiválasztjuk a Distributed File System szolgáltatást, alapértelmezésként mindkét alszolgáltatást is kijelöli számunkra a Server Manager, de ezután választhatjuk azt is, hogy nem kérjük őket, ha adott esetben nincs szükségünk rájuk.

## Képességek

A képességek olyan rendszerkomponensek, amelyek önmagukban nem határozzák meg a szerverünk hálózaton betöltött szerepét, tevékenységét. Valójában a képességek vagy támogató funkciót látnak el (vagyis lehetővé teszik a szerepkörök és szerepkör-szolgáltatások működését), vagy további extra, szerepkörhöz nem köthető funkciókat tesznek elérhetővé. Például a Failover Clustering egy képesség a Windows Server 2008-ban, és nem szerepkör – a Failover Clustering képesség ugyanis leginkább arra szolgál, hogy más szerepkörök rendelkezésre állását növeljük, például a File Server szerepkörét, illetve hogy szerepkörtől függetlenül biztosítsuk más erőforrások átterhelhetőségét.

Lássuk, melyek is ezek a képességek a Windows Server 2008-ban:

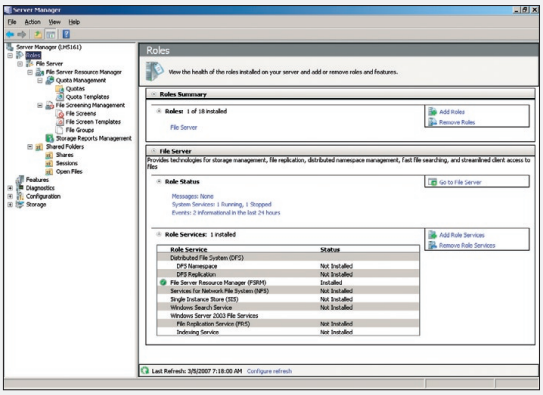
- .NET Framework 3.0
- BitLocker Drive Encryption
- BITS Server Extensions
- Connection Manager Administration Kit
- Desktop Experience
- Failover Clustering
- Group Policy Management
- Internet Printing Client
- Internet Storage Naming Server
- LRP Port Monitor
- Message Queuing
- Multipath I/O
- Network Load Balancing
- Peer Name Resolution Protocol
- Quality Windows Audio Video Experience
- Remote Assistance
- Remote Server Administration Tools
- Removable Storage Manager
- RPC over HTTP proxy
- Simple TCP/IP Services
- SMTP Server
- SNMP Services
- Storage Manager for SANs
- Subsystem for UNIX-based Applications
- Telnet Client; Telnet Server
- TFTP Client
- Windows Internal Database (SQL Server 2005 Embedded)
- Windows PowerShell
- Windows Process Activation Service
- Windows Recovery Disc
- Windows Server Backup
- Windows System Resource Manager
- WINS Server
- Wireless Networking.

Természetesen egyes szerepkörök és szerepkör-szolgáltatások telepítésekor szükség lehet néhány képesség telepítésére is, ezekre a Server Manager még a szerepkörök telepítése előtt fel is hívja a figyelmet, illetve felajánlja, hogy pótolja a hiányosságokat.

Érdekesség, hogy a Windows Server 2008 a Server Manager működéséhez a System Center termékek alapját képező és a Dinamikus Rendszerek Kezdeményezés egyik alapkövének számító System Definition Modelt használja fel.

E modellek segítségével képes eldönteni a Server Manager, hogy mik a komponensek közötti függőségek, és egyáltalán milyen telepítési lehetőségeink vannak. Ugyancsak érdekes, hogy az egyes komponensek valamennyi beállítási lehetőségét is pontosan definiálják ezek a modellek, így azokat akár parancssorból (a ServerManagerCmd.exe segítségével), akár a Server Manager felületéről is könnyen el tudjuk érni.

Látható, hogy a Server Manager és a mögötte húzódó infrastruktúra sokat segít majd abban, hogy szervereinket könnyebben tudjuk konfigurálni, és – hála az új felületnek – nemcsak szerepkörök telepítésére és eltávolítására, hanem a szerepkörök és képességek napi szintű kezelésére is ez lesz a legkényelmesebb felület a legtöbb rendszergazda számára.




*A Server Manager működés közben: minden beállítás egy helyen*

*Budai Péter*
(i-pbudai@microsoft.com) Microsoft Magyarország

# Windows Server 2008 AD – csak frappánsan

Kisebb és nagyobb változásokat egyaránt észrevehetünk majd a címtárszolgáltatások területén a következő generációs Windows-kiszolgálóban. A legjobb ezekben a változásokban az, hogy valódi, életszagú igényeket fednek le, illetve régóta elvárt, praktikus szolgáltatásokat valósítanak meg.

A címtárszolgáltatással kapcsolatos változások és fejlesztések minden új Windows-kiszolgáló esetén a fókuszba kerülnek. Nyilván nem véletlenül, hiszen a címtár-hierarchia rugalmassága és alkalmazhatósága miatt a tíz és a tízezer gépet tartalmazó hálózatok esetén is egyaránt jól használható az Active Directory, mégpedig a minden szervezet számára legfontosabb célra: a felhasználók, a számítógépek és egyéb erőforrások tárolására és kezelésére. Persze emellett a biztonsági „erőtér" megteremtése és egyéb fontos kiszolgálóalkalmazások, -megoldások (Exchange, Csoportházirend stb.) működésének támogatása is kritikusan fontos feladat. Az éppen a napokban aktuálissá vált Windows Server 2008 Beta 3 apropóján a következő újdonságokról, illetve változásokról számolhatunk be ebben a cikkben, a teljesség igénye nélkül:

- Read-Only tartományvezérlők (RODC);
- több tartományi jelszó- és kizárási házirend;
- az újraindítható címtárszolgáltatás;
- Snapshot Exposure.

## Read-Only tartományvezérlők (RODC)

Egy teljesen új tartományvezérlő-típusról van szó, amelyről elmondható, hogy ténylegesen valós igények hozták a felszínre. Egy mondatban összefoglalva: a RODC egy olyan DC, amely tartalmazza a címtár egy példányát, azaz képes az összes tartományvezérlő feladat ellátására, de a címtár tartalma nem változtatható meg helyben. Miért előnyös ez?

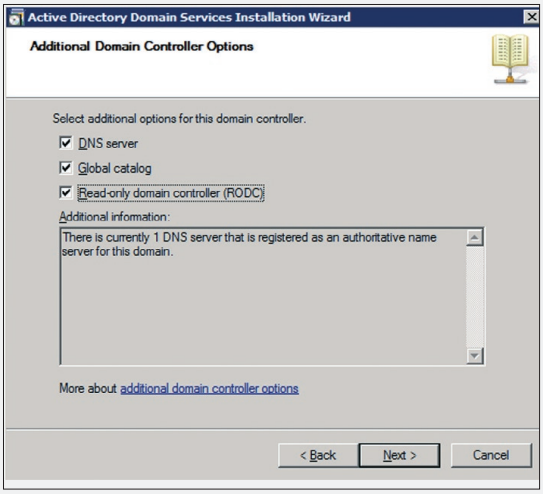
**Biztonságos.** Mivel nincs globális AD-módosítási lehetőség, olyan környezetbe is ajánlható, ahol fizikailag nem garantálható a biztonság, például egy védett szerverszobát nélkülöző telephelyre. Ha esetleg aztán az adott szervert helyben megpróbálják feltörni, vagy történetesen eltulajdonítják, az igazán szenzitív, globális tartományi adatokhoz nem lehet majd hozzáférni semmilyen módon, hiszen helyben ezek alapértelmezés szerint nem tárolódnak!

**Sávszélesség- és erőforrás-takarékos.** Mivel egy telephelyre biztonságosan telepíthető, a hitelesítési folyamatokhoz (például belépés, a helyi erőforrások elérése) nincs szükség a WAN-hálózatra vagy az internetre, igaz, ekkor némi kompromisszumra kényszerülünk, lásd később.

**Alkalmazás- és üzemeltetésbarát.** Előfordulhat, hogy az üzleti szempontból is fontos alkalmazások megkövetelik a tartományvezérlőt mint host gépet, vagy legalább a gyors elérését. Az is elképzelhető, hogy ezt az alkalmazást – más szerver híján – az egyetlen (telephelyi) szerverünkre kell feltelepíteni. Sőt, az is előfordulhat, hogy a speciális alkalmazást egy külső cég üzemelteti, azaz szüksége van interaktív belépésre, magas jogosultsági szinttel. Ki az, aki szívesen ad tartomány-rendszergazdai jogot egy ilyen esetben a külső szervezetnek? Viszont eddig – egy DC esetén – majdnem minden esetben muszáj volt, hiszen más lehetőségünk nem volt. Nos, a RODC esetén nyugodtabbak lehetünk, hiszen:

- nem lehet módosítani a címtár adott példányát helyben;

- tartományvezérlő mivolta ellenére van helyi Administrator csoport, azaz lehetséges az Active Directoryn kívüli minden mást




*A RODC szerepkör kiválasztása*

üzemeltetni egy nem Domain Admin felhasználói fiókkal!

**Üzemeltetésmentes.** Nincs AD-üzemeltetés, ergo nincs szükség magasabb szaktudású, Domain Admins jogosultsággal rendelkező szakemberre, hiszen nincs semmilyen, a tartományhoz, erdőhöz kapcsolódó üzemeltetési feladat.

Folytassuk a RODC megismerését a megoldandó technikai problémákkal, hiszen mivel teljesen új szerepkörről van szó, a garantált működéséhez szükség volt az ismert címtár és címtártámogató megoldások alapos és mélyreható korrekciójára.

## A Read-Only címtáradatbázis és a replikáció

A RODC „alatt” működő címtár-adatbázis-példánynak teljesen ugyanúgy kell kinéznie, mint egy hagyományos DC-nél (a jelszavakat kivéve, de erről majd később), mert különben hogyan megy le a replikáció, azaz hogyan lesz kompatibilis? Viszont változásokat nem tárolhat, és nem is replikálhat, azokat sem, amelyek esetleg szükségesek. Így aztán az összes változási kérelemnek el kell jutnia valahogyan egy írható DC-ig, hogy aztán a hagyományos replikációval visszakerülhessen a RODC-címtár példányába. Ez az a plusz kör, amely kizárja a korábbi telephelyi, sima DC esetén egyszerűen bevihető, és esetlegesen az egész erdőt negatívan érintő adatbázis-változásokat. Persze például a helyi alkalmazások továbbra is kaphatnak egyszerűen hozzáférést a címtár helyi példányához, de csak olvasási joggal. Ha ennél többre van szükség, akkor például LDAP-n keresztül automatikusan továbbkerül a kérés a hub site (a központi telephely vagy ahol egy írható DC van) felé.

Az úgynevezett unidirectional replikáció következménye az a változás is, hogy az írható DC-k a replikációs folyamatban felismerik: a partnerük egy Read-Only szerepkört tartalmaz, és ebben az esetben nem is kezdeményezik a „pull” típust, hiszen nem is jönne, nem is jöhetne semmilyen változás a RODC irányából. Ez megint csak sávszélesség-csökkentést jelent, és egyúttal a hídfő-DC-ket sem terheljük annyira. Ide tartozik az is, hogy ez az új, egyoldalas replikáció nemcsak a címtárszolgáltatásoknál jelentkezik, hanem értelemszerűen az ugyanígy használható DFS-R-nél is.

## A hitelesítési adatok gyorsítótárazása

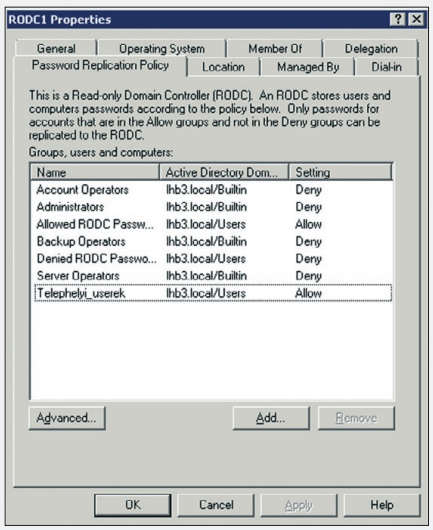
Alapértelmezés szerint a RODC – két kivételtől eltekintve – nem tartalmazza semmilyen felhasználói vagy számítógépfiók jelszavát. E két kivétel a RODC gépfiókja, illetve a speciális szerepet betöltő krbtgt fiók. Viszont arra van lehetőségünk, hogy bármely más fiók hitelesítési adatait gyorsítótárazzuk. Miért akarnánk ezt tenni? Egyszerű: ne kelljen „kimenni” az adott hálózatból, az esetlegesen lassú kapcsolaton keresztül egy központi DC-hez, mondjuk, minden felhasználói belépés vagy egyéb hitelesítés esetén. De hogyan lehetséges az ilyen típusú adat eltárolása és kiszolgálása?

A RODC képes KDC-ként (Key Distribution Center) viselkedni a telephely felhasználóival és gépeivel szemben, azaz képes lesz tökéletes és érvényes Kerberos-kulcsokat kiadni, amelyeket aztán a fiókok teljes körűen használhatnak is a hitelesítési folyamatban – a központi DC-k nélkül is. Viszont már most tudnunk kell, hogy a RODC a TGT-kérések aláírásához és titkosításához a saját krbtgt fiókját és annak jelszavát is használhatja, amely nyilván különbözni fog egy központi DC ugyanerre a célra használt krbtgt fiókjának hitelesítési adataitól. A folyamatban e két forrás különbözősége lesz a kulcs.

Szóval, az első alkalommal a hitelesítés biztosan csak egy központi DC-vel fog menni, mert ugye a kliens az alapértelmezés szerint szeretné használni a RODC-t, de ekkor ez még csak továbbítani tudja a kérését. Ha ezzel a közvetítéssel sikerül a hitelesítés, akkor a RODC el fogja kérni az adott fiók hitelesítési adatait.

Persze, az írható DC csak úgy nem adja oda, hanem – miután felismerte, hogy ez egy RODC kérése – az úgynevezett Password Replication Policy (PRP) alapján dönti el, hogy szabad-e ezt tennie vagy sem. A PRP gyakorlatilag egy mini táblázat, amelybe manuálisan kell felvennünk azokat a csoportokat, gép- vagy felhasználói fiókokat, amelyekről úgy ítéljük meg, hogy hitelesítési adataikat nyugodt szívvel merjük gyorsítótárazni a RODC-n.

Alapesetben ez a táblázat teljesen üres, azaz minden fiók és csoport számára tiltott ez a




*A PRP fül a RODC tulajdonságai között az ADUC-ban*

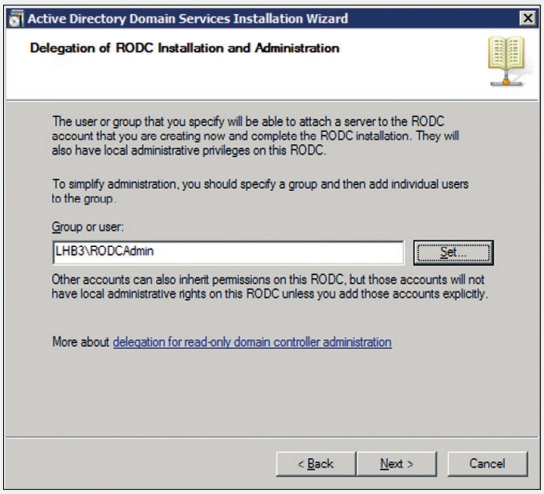
lehetőség. Ha viszont a kérdéses fiók számára engedélyezve van a gyorsítótárazás, akkor a központi DC átnyújtja a megfelelő adatokat a RODC-nek, amely meg szépen elraktározza ezeket, majd – most jön a helyi krbtgt fiók szerepe – a saját fiókjával aláírt TGT-t adja oda a kliensnek.

## A második belépés

De mi történik a második belépéskor? Merthogy immár van tárolt jelszó helyben is, azaz a RODC képes lenne ellátni a hitelesítést

közvetlenül is, de honnan tudja, hogy ezt megteheti? Egyszerű: a RODC képes felfedezni az adott TGT-n a saját krbtgt fiókjának nyomát, vagyis ha megtalálja, akkor automatikusan nem küldi tovább a központi DC felé a kérést, hanem a helyben tárolt adatokkal gyorsan és problémamentesen megoldja a hitelesítést.

A Password Replication Policy „feltöltése" abszolút a mi döntésünk – mérlegelnünk kell tehát, hogy mely fiókok vagy csoportok azok, amelyeknek a hitelesítési adatai lekerülhetnek a RODC-re. Ha minden, a telephelyen használt fiókot engedélyezünk (és esetleg a tartományi admin-fiókokat is!), gyors lesz a belépés, viszont ha eltulajdonítják a gépet, hozzáférhetőek a jelszavak, ugyanúgy, mint egy hagyományos DC esetén. Ha csak néhány szimpla felhasználói fiókot engedélyezünk, akkor több idő megy el más fiókok esetén a belépésre, viszont nincs komoly biztonsági probléma.




*A RODC helyi admin-fiókjának kijelölése a telepítés során*

## Az admin-jogok szeparálása

Már említettük, hogy a RODC-n szükséges és fontos is egy helyi, magas szintű jogosultság biztosítása, ami hozzávetőleg a lokális admin jogkörével egyenlő – anélkül, hogy a címtár objektumaira bármilyen befolyása lenne az ebbe a csoportba tartozó felhasználóknak.

Egy ilyen fiók csak egy tartományi fiók lehet (célszerűen az adott telephely egy felhasználója), és ami még fontos, ha egy másik helyszínen, egy hagyományos tartományvezérlőn lépne be ez a felhasználó, akkor ez ugyanúgy nem fog sikerülni neki, mint mielőtt megkapta volna ezt a lehetőséget a RODC-n. Egy fiók e csoportba történő belehelyezése egyébként kétféle módon történhet meg:

1. a parancssorból a Dsmgmt eszközzel;
2. a RODC telepítése során a varázsló egyik lépéseként.

## Read-Only DNS

„Ha DC, akkor DNS-szerver is". Ezt a tételt a RODC esetén is tudjuk érvényesíteni. A RODC DNS-szerver teljes értékű, például képes az összes, a DNS által használt alkalmazáspartíció replikálására (ForestDNSZones, DomainDNSZones) vagy a kliensek maradéktalan névfeloldási kéréseinek kiszolgálására. De... A RODC jellegéből adódóan nem minden művelet történhet meg. Melyek ezek? Nos, ide tartozik például a kliensek automatikus regisztrációja a DDNS segítségével vagy saját maga felvétele például egy AD-integrált zónába, egy NS-rekord alá.

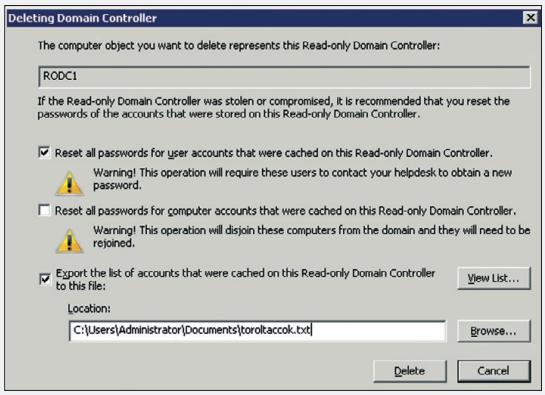
Így aztán, ha egy kliensgép saját rekordjának frissítését végezné el, akkor a RODC DNS közli vele, hogy mely DNS-szerveren teheti ezt meg, merthogy helyben szó sem lehet róla.

Mindeközben azért – a háttérben – megkísérli a megfelelő DNS-szerverről lehúzni a kliensre vonatkozó változást, azért, hogy a következő pillanatban már ki tudja szolgálni egy másik kérés során ezt a nevet/címet. Ugyancsak fontos, hogy szerencsére ez a replikáció csak az adott DNS-rekordra vonatkozik, nem kell tehát egy egész zónát „lehúzni" a folyamat során.

## A RODC bevezetésének feltételei

Nem kevés, „súlyos" elem van ebben a listában, de talán az eddigiek alapján látszik, hogy valóban mélyen bele kellett nyúlni a címtár működésébe a RODC-k bevezetése miatt:

- A PDC Emulator FSMO szerepét betöltő DC csak Windows Server 2008 lehet. Ez elsősorban a RODC krbtgt fiókjának „legyártásához", illetve egyéb RODC-műveletekhez szükséges.
- Az a DC, amelyhez a RODC a hitelesítési kéréseket intézi majd, csak Windows Server 2008 lehet, hiszen a Password Replication Policy csak az új szerverrel képes működni, illetve felismerni, hogy egy olyan speciális kérésről van szó, amelyet egy RODC adott ki.
- A tartomány működési szintjének legalább Windows Server 2003-nak kell lennie, azért, hogy elérhetővé (azaz inkább kikényszeríthetővé) váljon a biztonságos Kerberos-delegálás.
- Az erdő működési szintjének tekintetében is kötelező a minimum Windows Server 2003-as szint, az úgynevezett linked-value replikáció használata miatt, amely nagyobb replikációs megbízhatóságot nyújt, illetve lehetővé teszi, hogy ne az adott elemet tartalmazó egész tömb replikálódjon, hanem csak a ténylegesen megváltozott elem.
- A Password Replication Policy használatának alapfeltétele a sémabővítés.
- Használnunk kell az Adprep/rodcprep parancsot az erdő szintjén, ami azért szükséges, hogy frissítsük az erdő összes DNS-alkalmazáspartícióját, hogy aztán az összes RODC DNS-szerver képes legyen – immár a megfelelő jogosultsággal – replikálni ezeket a rekordokat.




*A RODC manuális eltávolítása a címtárból*

## A RODC eltávolítása

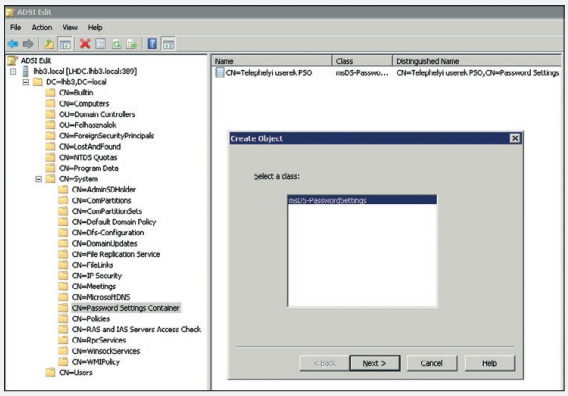
Ebben a témakörben is van némi praktikus változás, azaz a RODC törlésekor kapunk segítséget ahhoz, hogy gyorsan orvosoljuk az eltulajdonítás vagy valamely drasztikus változás okozta károkat.

Egy ennek megfelelő esetben a törlés előtt (a következő ábrán jól látható módon) a

RODC által is tárolt hitelesítési adatokat lenullázhatjuk.

## Több tartományi jelszó- és kizárási házirend

Jelen állás szerint egy Windows Server 2003-tartományban semmilyen megoldást nem találhatunk az egy tartomány = egy jelszóházirend tétel kikerülésére. Ha valamilyen nyomatékos okból mégis muszáj egy új jelszóházirendet definiálni, akkor csak egyet lehet javasolni: egy új tartományt kell létrehozni, ami persze nem tökéletes megoldás, talán több is a hátránya, mint az előnye. A WS08-ban végre megszűnt ez a korlát, egy teljesen új módszerrel – talán kicsit nehézkesen, de – kreálhatunk azonos tartományban több jelszóházirendet is, sőt az új házirend kiterjed a fiókkizárási (account lockout) opciókra is.



*Egy PSO már kész van, következhet a második*

Miért fontos a több tartományi jelszóházirend? Nos, ez eléggé értelemszerű, hiszen mivel a felhasználói fiókok „súlya” nem azonos, a magas jogosultságú fiókokat jobban kell(ene) védenünk, erősebb jelszavakat lenne célszerű megkövetelnünk, azért, hogy az emberi tényező (hanyagság, felületesség, felelőtlenség) által okozott problémákat megelőzzük. Emellett a normál felhasználói fiókok jelszavával kapcsolatban nem minden esetben szükséges kőkemény restrikciókat alkalmazni, nem indokolt az átlagos felhasználókat „kínozni” az extra jelszómegadási kritériumokkal.

Egy alternatív jelszó- és kizárási házirend létrehozásának lépései három pontban foglalhatók össze:

1. Készítsük el a megfelelő csoportot és mozgassuk át a megfelelő fiókokat!

2. Készítsük el az új PSO-t (Password Settings Objects), azaz az új jelszóházirendet!

3. Rendeljük hozzá a PSO-t az adott csoport(ok)hoz vagy akár egyesével a felhasználói fiók(ok)hoz!

Ebből már kiderülhetett, hogy az új jelszóházirendeket csak fiókokhoz vagy globális biztonsági csoportokhoz rendelhetjük. Mi lesz az OU-kkal? Nos, sajnos közvetlenül nem rendelhető hozzá egy PSO egy OU-hoz, ha maradunk ennél a hierarchiánál, akkor muszáj legyártani az „árnyék” biztonsági csoportokat. Ez kissé bonyolítja talán a folyamatot, de gondoljunk bele, mennyi csoportunk van viszont már készen, gyárilag létrehozva (Domain Admins, Enterprise Admins, Schema Admins, Server Operators, Backup Operators stb.)?

A PSO-k létrehozása egyébként kétféle módon történhet, ADSI Edittel (adsiedit.msc) vagy ldifide-vel. Az első módszer első lépése a következő útvonalon egy új objektum létrehozása:

```
<domain_name>,CN=System,CN=Password
Settings, CN=Password Settings Container
```

Az ezután következő, tíznél is több lépést tartalmazó varázsló beállításai között ráismerhetünk a szokásos jelszó-, illetve kizárási házirend opciókra. Időnként kissé bonyolultabb a mezők kitöltése, például csak másodpercben lehetséges értéket megadni, vagy az érvényesítési területet csak a distinguishedName értékkel (CN=,DC= stb.) lehetséges kijelölni. Ha a varázslót végiglépkedtük, akkor készen van az új házirend, és már érvényre is jutott. Innentől viszont nem kell az ADSI Edit az esetleges korrekcióhoz, az ADUC-ban a System\Password Settings Container alatt megtaláljuk (feltéve, ha engedélyeztük a View alatt a Advanced nézet opciót) és szerkeszthetjük az alternatív házirendeket. Hogyan? A szintén teljesen új (és szinte minden objektumnál elérhető) Attribute Editor fül segítségével.

## Kritériumok és további megjegyzések

1. Először is az egész folyamat csak akkor indítható el, ha az adott tartomány domain-funkcionalitási szintje Windows Server 2008, és megtörtént a sémabővítés is (két teljesen új osztállyal kell kibővíteni a sémát). Ez viszont csak akkor érhető el, ha már likvidáltuk az összes Windows 2000/2003 Server DC-t.

2. Csak a Domain Admins csoport tagjai készíthetnek és alkalmazhatnak PSO-kat a fiókokra vagy a csoportokra. Olvasási jogot szabadon delegálhatunk a PSO-ra, de egy viszonylag életszerű példát említve, egy helpdeskes kolléga nem fogja tudni megváltoztatni a jelszóházirendet.

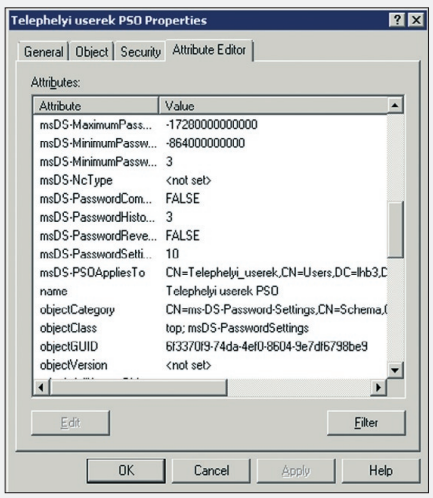
3. Számítógép-fiókokra semmilyen körülmények között nem alkalmazhatók az új jelszó- és kizárási házirendek.

4. A testre szabott jelszó-filterekkel már szerencsésebbek vagyunk, mert minden további következmény nélkül használhatjuk ezeket továbbra is.

## Az újraindítható címtárszolgáltatás

A Windows Server 2008 tartományvezérlőkön a címtár újraindítható. Miért? És hogyan?

Elsősorban azért, hogy ne kelljen újraindítani a gépet bizonyos esetekben, például a címtárt érintő frissítések vagy éppen az AD karbantartása (például offline defrag) apropóján. Meg aztán, amíg tart az újraindítás – ez általában, szinte függetlenül a gép teljesítményétől rengeteg idő –, ne essenek ki egyéb, a tartományvezérlőn futó kritikus szolgáltatások, például a DHCP-szerver „csont nélkül” működik majd tovább. A címtárszervizek leállítása és újraindítása bárme-



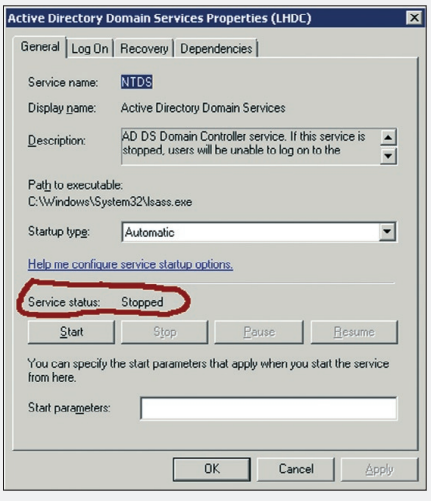

*Egy kész PSO utólagos konfigurációja az ADUC-ból*

lyik WS08-tartományvezérlőn lehetséges, és nincs semmilyen egyéb megkötés sem, azaz az eddigi általános helyzettel szemben szó sincs például arról, hogy ez a lehetőség funkcionalitásiszint-függő lenne.

Az újraindítási opció minimális változást hoz a kezelésben, és nincsenek extra opciók sem ezzel kapcsolatban, azaz tényleg csak annyiról van szó, hogy a DC-ken lévő Services MMC-ben megjelenik a listában a Domain Controller nevű szerviz, amelyet a szokásos módon lehet kezelni. Az AD ily módon leállított állapotára egy külön, fantázia nélküli kifejezés van, úgy hívják: „AD DS Stopped" üzemmód.

Igazából talán inkább az az érdekes, hogy ilyenkor mi történik a szerverrel a tartományban? Vagy újra lehet használni a helyi felhasználó-adatbázist? Ezt azért nem. Tag marad erre az időre egyáltalán a tartományban? Vagy tagkiszolgáló? Vagy egyik sem? Nos, ha egyedül van a tartományban, akkor vélhetően logikus, hogy egyik sem. Viszont ha több DC is van, akkor a tartományi tagsága él, és tagkiszolgálóként dolgozik addig is, amíg újra DC nem lesz. Így tehát például az interaktív vagy a hálózaton keresztüli bejelentkezés lehetősége ebben az esetben is adott. Valamennyire akkor is igaz ez, ha nincs elérhető másik DC, mert ekkor a helyi belépéshez a Directory Services Restore jelszót kell használnunk.

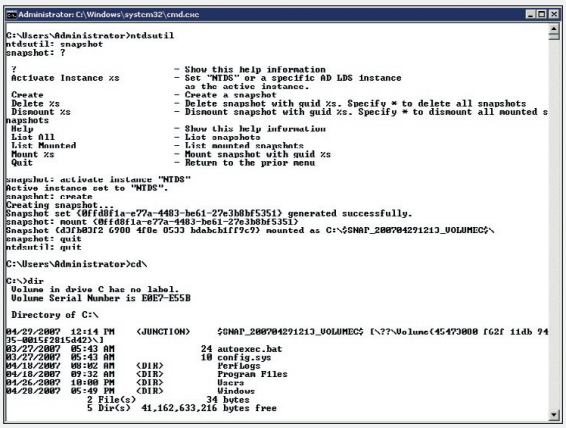
Sokáig persze nem célszerű azért így hagyni a gépet, hiszen a beléptetés vagy a replikáció természetesen nem működik, az adatbázis (Ntds.dit) offline, és a szerviz leállítása értelemszerűen magával húz a sötétségbe más szolgáltatásokat is (DNS, KDC, FRS stb.). Ellenben ilyenkor a speciális Directory Services Restore üzemmód továbbra is minden korlátozás nélkül használható, hiszen ehhez nem kell senkihez sem fordulnia az árva DC-nek – a jelszó érvényesítése, azaz az effajta bejelentkezés a szokásos módon, helyben megtörténhet.




*Egy leállított Active Directory-szolgáltatás*

## Snapshot Exposure

Ismét egy teljesen új megoldásról van szó, amely – tömören – abban segíti az üzemeltetőket, hogy egyszerűen azonosítsuk azokat a címtárobjektumokat, amelyeket így vagy úgy, de töröltünk vagy éppen megváltoztattunk. Bár visszaállítani nem fogjuk tudni ezzel a módszerrel, de mielőtt nekiesnénk a tényleges visszaállításnak, gyorsan áttekinthetjük, hogy mit kell és mit lehet majd visszahozni. A legfontosabb viszont, hogy ezeket a „pillanatfelvételeket" vagy mentési példányokat anélkül tudjuk megtekinteni, hogy a speciális AD Restore Mode miatt újra kellene indítani a gépet.




*Egy pillanatfelvétel készítése, mountolása és kilistázása*

A megvalósítás egyes lépései igényelnek némi szakértelmet, és részben parancssorból történnek.

1. Indítsuk az Ntdsutil.exe-t, és használjuk az új „snapshot" parancsot, amellyel készíthetünk egy mentést az AD-ról, majd ezt fel is csatolhatjuk a fájlrendszerbe.

2. Egy másik parancssori eszköz jön, a Dsamain.exe (Exchange-örökség), amellyel az adott példányt LDAP-szerverként tudjuk futtatni. A szintaxisra vigyázzunk, és persze arra is, hogy a kötelezően mellékelendő 4 port (LDAP, LDAP-SSL, GC, GC-SSL) mindegyike eltérő legyen a szokásostól, azaz bármi lehet, csak ne a szabvány, hiszen a működő AD ezeket használja.

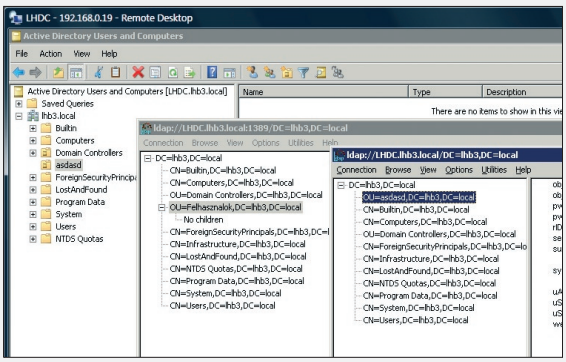
3. Futtassuk az ldp.exe-t a szokásos módon, de ne a szokásos porton. Az LDAP-port az legyen, amit az előző pontban megadtunk.

4. Kész, immár online tallózhatjuk az előző AD-verziót!

5. Ha végeztünk, a Dsamain.exe ablakában állítsuk le az AD mentett példányát a CTRL+C-vel.

6. Az Ntdsutil-t úgy is konfigurálhatjuk, hogy rendszeres időközönként megtegye az automatikus pillanatfelvétel-készítést, így aztán valóban bármikor visszanézhetünk majd a régebbi példányokba is.

Egyetlen fontos dolog maradt még ezzel az újdonsággal kapcsolatban, amire nagyon oda kell figyelnünk, ez pedig a biztonság. Alapesetben csak a Domain/Enterprise Admin csoport tekintheti meg a pillanatfelvételeket, de sajna bármelyik erdőből! Mindez azt jelenti, hogy ha valaki átmásolja a fájlrendszerből a pillanatfelvételt egy másik erdőbe, ahol történetesen Domain Admin, akkor minden további nélkül belenézhet a mi címtárunkba. Ezért ezeknek a példányoknak a biztonságáról feltétlenül érdemes valamilyen egyéb módszerrel külön is gondoskodni.




*Két AD-példány egyszerre elérhető az ldp.exe-vel, és látszik a különbség is, mivel a pillanatfelvétel készítése után átneveztük az egyik csoportot!*

*Gál Tamás*
(v-tagal@microsoft.com) Microsoft Magyarország

# Maximum és minimum – a Server Core

Négyszáz megahertzes CPU? 128 megabájt RAM egy teljes értékű DC esetén? 5-6 gigabájt helyfoglalás egy olyan Windows-kiszolgálónak, amelyik jelen pillanatban 23 különböző kiszolgálószerepet képes ellátni? Alapesetben kevesebb, mint 50 automatikusan induló rendszerszerviz? Nem ment el az eszünk, és nem a Windows NT 1.0-ról van szó. Ezek a hardverkritériumok a Windows Server 2008 egy teljesen új típusú verziójára érvényesek.

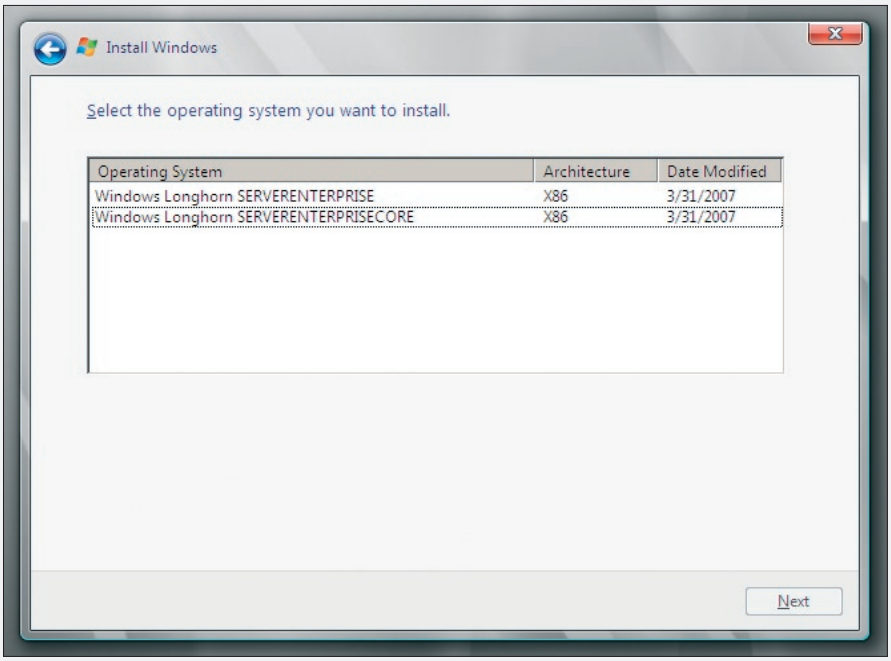
A Windows Server 2008-család a szokásos Standard, Enterprise és egyéb verziók mellett egy különleges változatot is tartalmaz majd, amelynek neve jelenleg (a cikk írásakor még közvetlenül a Beta 3 előtti állunk): Server Core. Különlegessége elsősorban abból áll, hogy 95 százalékban parancssorból működik, azaz egyáltalán nincs GUI. Elsőre ez biztosan meghökkentőnek tűnik, de működik, és nem is akárhogyan.

## Előnyök és hátrányok

Nézzük sorban, milyen előnyei vannak egy ilyen kiszolgálóverziónak:

- Az erőforrások szempontjából lényegesen gyengébb gépen is jól működik. A tesztjeink során kiderült például, hogy egy virtuális gépben 80 megabájt RAM-mal már egy kényelmesen felszerelt tagkiszolgáló is működtethető, 128 megabájt memóriával pedig egy full extrás tartományvezérlő is tökéletesen jól teljesít. Ha igazi vasról van szó, hasonló a helyzet, annak ellenére, hogy – szintén a tapasztalatok alapján – ilyenkor kicsit több erő kell. De nem sokkal, az ajánlás szerint 256 megabájt RAM elég a teljes funkcionalitáshoz. Ide tartozik a lemezhelyigény is, ami az alaptelepítés után a pagefile nélkül valóban nem több, mint 6 gigabájt (és ebben benne van a majdnem 3 gigabájtnyi, kompatibilitási okokból az alkalmazások számára fenntartott Windows\Winsxs mappa tartalma is).
- Számos kiszolgáló-feladatkört képes ellátni a Server Core (erről később), de lényegesen kevesebbet, mint például egy tipikus Windows Server 2008. Ezenkívül nem lehet akármit rátelepíteni, azaz léteznek a belső és a harmadik gyártótól származó programok területén is kemény korlátok. A „kevesebb jobb” elv alapján ez nyilván sok környezetben nagyon fontos lesz, hiszen itt szintén nem kevés üzemeltetési időt takaríthatunk meg.
- Becslések szerint kb. 60 százalékkal kevesebbet kell a biztonsági és egyéb javításokkal törődnünk, ha nincs GUI, és nincs az ehhez szorosan kötődő rengeteg alkalmazás. Ez nyilván azt is jelenti, hogy kevesebbet kell ezzel a kiszolgálóval foglalkozni a beüzemelés után („...ott lehet hagyni a sarokban”), és nincs annyi újraindítás.
- A telepítés utáni indító konfigurálás (például TCP/IP, gépnév megváltoztatása stb.) mindenképpen a parancssorból történik, de ezután minimum háromféle módszerrel vagyunk képesek távolból is felügyelni, üzemeltetni a Server Core-t.

  Használhatjuk az MMC-t, az RDP-t és a Vistában, W2K3 R2-ben meglévő WS-Management képességet, azaz a WinRM/WinRS párost, ami gyakorlatilag távoli parancssorként működik. Tehát nem kell




*Itt dől el minden*

halálra rémülni a szerverkonzolon a fekete háttér előtt villogó fehér kurzortól, léteznek módszerek a mindennapok feladatainak elvégzésére például a rendszergazda gépéről is.

- Minden WS08-verzióban (Standard, Enterprise) megtalálható lesz, és egyformán használható x86/x64-környezetben is.

A teljesség és a tisztánlátás kedvéért tekintsük át a hátrányokat is, mert azért sejthető, hogy a felsorolt előnyök számos kompromisszummal is járnak.

- Tényleg nincs GUI. Nincs Explorer, MMC, CLR, Shell, IE, Media Player, OE, RDP kliens stb. El kell gondolkodnunk azon, hogy hogyan lehet DNS-zónát telepíteni parancssorból? Hogyan lehet szintén innen felhasználót felvenni az AD-ba? Hogy csinálunk egy kivételszabályt a tűzfalban, hogyan hitelesítünk egy DHCP-szervert az AD segítségével, ha nincs GUI? Még sok ilyen kérdést fel fogunk tenni magunknak, de a válasz végül mindig az, hogy lehet, csak kicsit (ritkán nagyon) bonyolultabb.
- Ami előny, az egyben hátrány is, azaz kevesebb komponens és alkalmazás működik a Server Core-kiszolgálókon. Hét fő szerepkör van, amelyeket ellát(hat): DHCP, DNS-kiszolgáló, fájlszerver, Active Directory, Active Directory Lightweight Directory Services (AD LDS, korábban ADAM), Print Server és Media Services.

Ezek mellett azért van egy tekintélyes listánk az egyéb szerepekről is:

- BitLocker és BitLocker Remote Admin Tool;
- Client for NFS;
- DFS Server, DFS Replication;
- Failover Cluster;
- FRS;
- MultipathIO;
- Removable Storage Management;
- Network Load Balancing;
- LPD Print Service;
- NFS Server, Subsystem for UNIX-based Applications;
- QoS (Qwave);
- Single Instance Storage;
- SNMP;
- Telnet Client;
- Windows Server Backup;
- WINS.

Le kell szögezni még egyszer, hogy ez a Beta 3 körüli állapot, azaz még változhat, például 2006 novembere óta kétszer is bővült a szolgáltatáscsomag, nem is kevés elemmel.

Némi hátrány mutatkozik a telepítés, pontosabban a frissítés és migrálás környékén is. Három fontos részletről van szó:

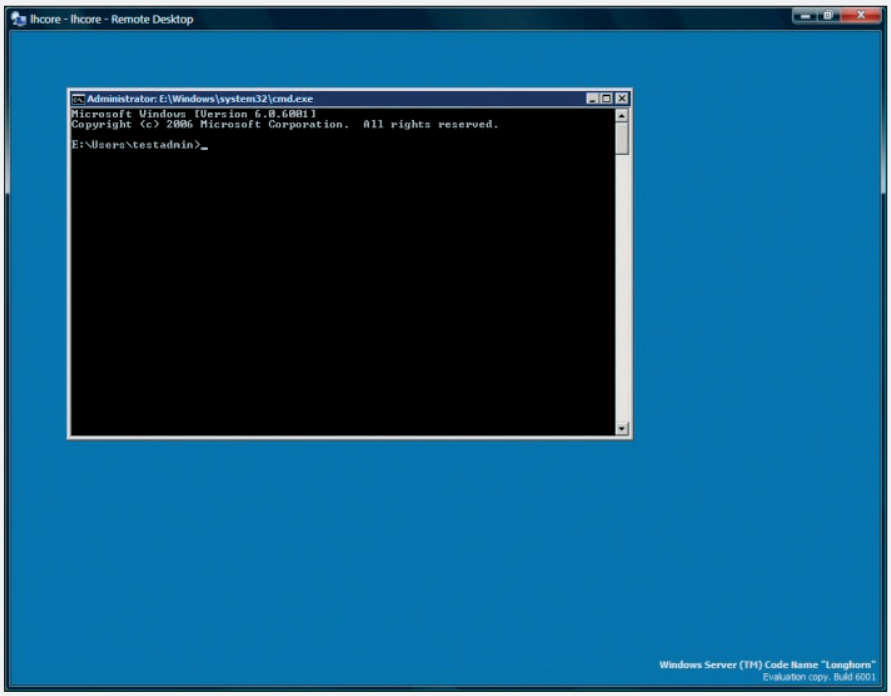
1. Nem lehetséges frissíteni egy korábbi Windows-szerververzióról.

2. Nem járható út a „nagy” Windows Server 2008-verziókról történő frissítés sem.

3. A Server Core-t szintén nem lehet a „nagy” Windows Server 2008-ra frissíteni.

Ezekből értelemszerűen az következik, hogy a Server Core-telepítés csak tiszta (clean) telepítés lehet.

Egy komoly előny viszont azonnal látszik a telepítésnél, ugyanis villámgyors, hozzávetőleg 15 perc, és kész vagyunk. Még egy fontos dolog: a csendes telepítés megvalósítható, a




*A Server Core-ból ennyi látszik a belépés után*

Server Core egy unattend.xml alapján képes települni, azaz testre szabhatjuk (képernyőfelbontás, RDP engedélyezése stb.), valamint automatizálhatjuk a telepítést például a BDD 2007-tel vagy önmagában (a WAIK részeként működő) a Windows System Image Managerrel.

## Első lépések

A telepítésben semmi extra nincs, a Vistánál és a Windows Server 2008-nál is tapasztalható módon alig kell hozzányúlni (a termékkulcs ugyanaz lesz, mint a nagy WS08-nál).

Ha kész, az újfajta egész képernyős belépési képernyőt láthatjuk. Tudnunk kell, hogy egyetlen működő felhasználói fiók van csak (ez az Administrator, persze van még egy, a Guest, de szokás szerint letiltva), és ennek sincs még jelszava. Ha belépünk, a lenti (néhányunk számára elsőre valószínűleg lelombozó) látvány tárul a szemünk elé.

Az első teendők közé tartozik tehát az admin-jelszó megadása, amelyhez a legkézenfekvőbb módszer a CTRL+ALT+DEL, és ekkor azért némi vigasz gyanánt kaphatunk valamilyen grafikus felületet, ahol a jelszóváltoztatáson kívül ki is léphetünk, vagy lezárhatjuk a gépet, illetve elindíthatjuk a Task Managert is. A gép újraindításához és leállításához is ide vezet az utunk.

Egyébként csak a teljesség kedvéért a jelszóváltoztatáshoz a *net user administrator* * parancs is megfelelő. De vajon mi a következő lépés? Eltaláltuk: a TCP/IP bekonfigurálása. Persze ha van DHCP, és megfelel nekünk, akkor nincs probléma, de kiszolgálóknál ez nem így szokás, úgyhogy jöhet a manuális beállítás, de először tájékozódjunk:

```
netsh interface ipv4 show interfaces
```

Ez azért is különösen fontos most, mert az eredményből több adatot is hasznosítani fogunk a későbbiekben, például az adott hálózati kapcsolat pontos nevét, valamint a sorszámát.

Ezután jöhet a tényleges konfigurálás:

```
netsh interface ipv4 set address name=2 source=static address=x.x.x.x mask=x.x.x.x gateway=x.x.x.x
```

(A Name utáni sorszám a hálózati kapcsolat sorszáma az előbbi listából az Idx oszlop alól.)

Persze, vissza is állíthatjuk bármikor a DHCP-t:

```
netsh interface ipv4 set address name=2 source=dhcp
```

A DNS-kiszolgáló beállítása kulcsfontosságú feladat:

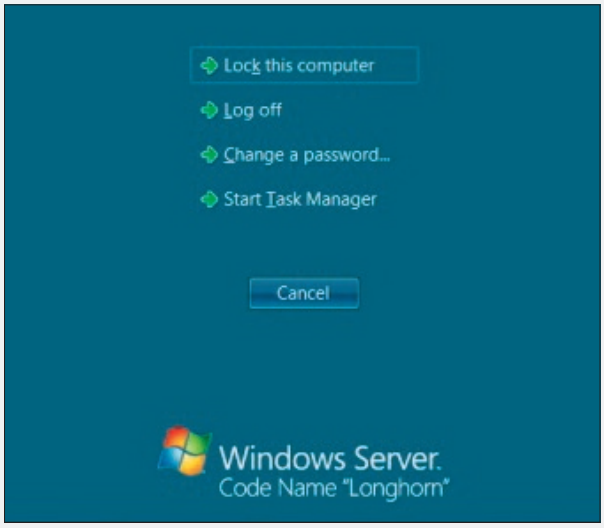
```
netsh interface ipv4 add dnsserver name=2 address=x.x.x.x index=1
```

(Mivel több DNS-szerver is felvehető, az index adja meg a használandó DNS-szerverek sorrendjét.)

A telepítés közben a szokásos véletlenszerűen kiválasztott, hiperérthetetlen nevet kapja a gép, ebbe a folyamatba közben nem avatkozhatunk bele, utólag viszont igen, mégpedig az ismerős netdom paranccsal:

```
netdom renamecomputer GepMostaniNeve /newname:GepUjNeve
```

Felmerülhet a kérdés: hogyan derítjük ki a gép jelenlegi nevét. Nos, a legelső alkalommal utána kellett nézni, de aztán kiderült, hogy a „hostname” parancs működik itt is, sőt a „set c” és a „systeminfo” is.




*A mini Start menü (majdnem ugyanaz mint a Vistánál)*

Ha majd letöltjük a Beta 3 publikus verzióját (a TechNet Magazin e számának megjelenésekor már valószínűleg lesz ilyen), akkor azzal is szembesülni fogunk, hogy aktiválásra szorul. Erre mostanság nem is kapunk túlságosan sok időt, a Windows Server 2008-nál például 3 napunk van, szóval tartozzon ez is az első lépések közé, mert „késő bánat, eb gondolat”. A szükséges parancs:

```
Cscript c:\windows\system32\slmgr.vbs -ato
```

Ha kiadjuk, hozzávetőleg 1-2 percig nem történik az égvilágon semmi látható, majd ezután diszkréten közli egy apró panelen, hogy sikerült. Egyébként az aktiválás állapotának kiderítéséhez a következő parancsra lesz szükség:

```
Cscript c:\windows\system32\slmgr.vbs -xpr
```

Mint szinte minden lépésnél, itt is van lehetőség távoli végrehajtásra, egy másik gépről:

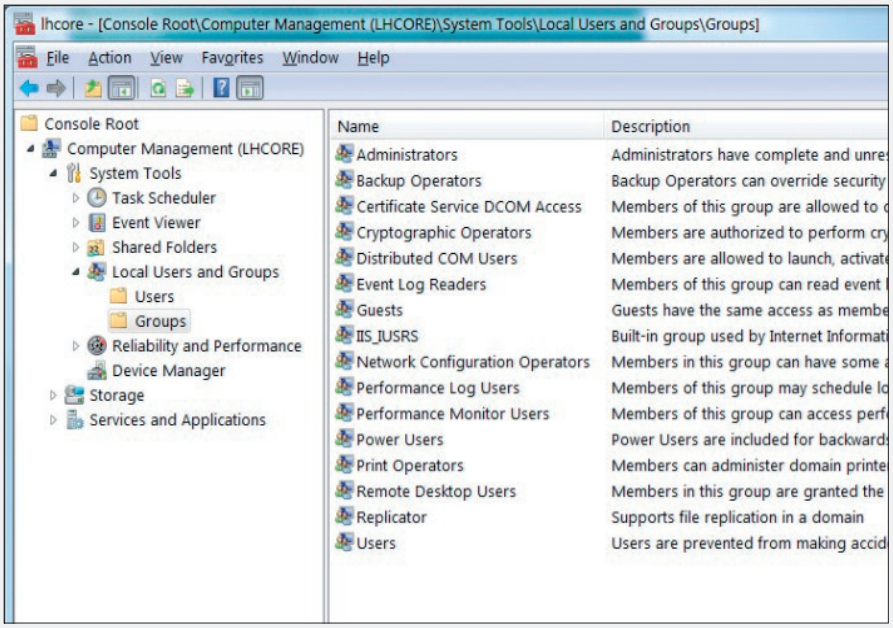
```
Cscript c:\windows\system32\slmgr.vbs gépneve\administrator jelszó -ato
```

Ha nem a Server Core lesz a tartományunk alapköve, hanem egy létezőbe szeretnénk beléptetni, akkor már az elején célszerű gondoskodni erről, mivel a felügyelet (például a WinRM) is feltétele ennek, vagy ha nem, akkor is sokkal egyszerűbb a megoldás (például az MMC). Ehhez gépeljük be a következő parancsot:

```
netdom join gépnév /domain:domain_név /userd:user_neve /passwordd:*
```

Ennyi. Nincs újraindítás, röpke pár másodperc után a gép a tartomány tagja. A user_neve természetesen egy olyan felhasználói fiók, amelynek van megfelelő jogosultsága beléptetni a gépet a tartományba, a passwordd* pedig nem elírás, a csillag hatására kéri be a jelszót.

Természetesen a Domain Admins csoport automatikusan tagja lesz a helyi Administra-




*A csoportokat tekintve nincs sok különbség a „nagy" Windows Server 2008-hoz képest*

tors csoportnak a tartományba léptetés után, de ha mégis szükségünk lesz egy tartományi felhasználó helyi admin csoportba helyezésére, használjuk ezt a parancsot:

```
Net localgroup administrators /add domain_neve\
user_neve
```

## Ellenőrzés és felügyelet

Mint ahogyan már említettük, a felügyelet elláthatóvá távolból három különböző módszerrel is, és igazából célszerű is ez, hiszen helyi eszköz viszonylag kevés van.

A három módszer közül az egyik az RDP-kapcsolat, amelyet először engedélyezni kell a kiszolgálón:

```
cscript C:\Windows\System32\Scregedit.wsf /ar 0
```

De van itt egy kis trükk is, mert ez az engedélyezés az RDP 6.0-s kliensekre vonatkozik csak (Vistán alapból ez van, XPSP2-re letölthető), ha régebbi RDP-kliensről óhajtjuk kezelni, akkor:

```
cscript C:\Windows\System32\Scregedit.wsf /cs 0
```

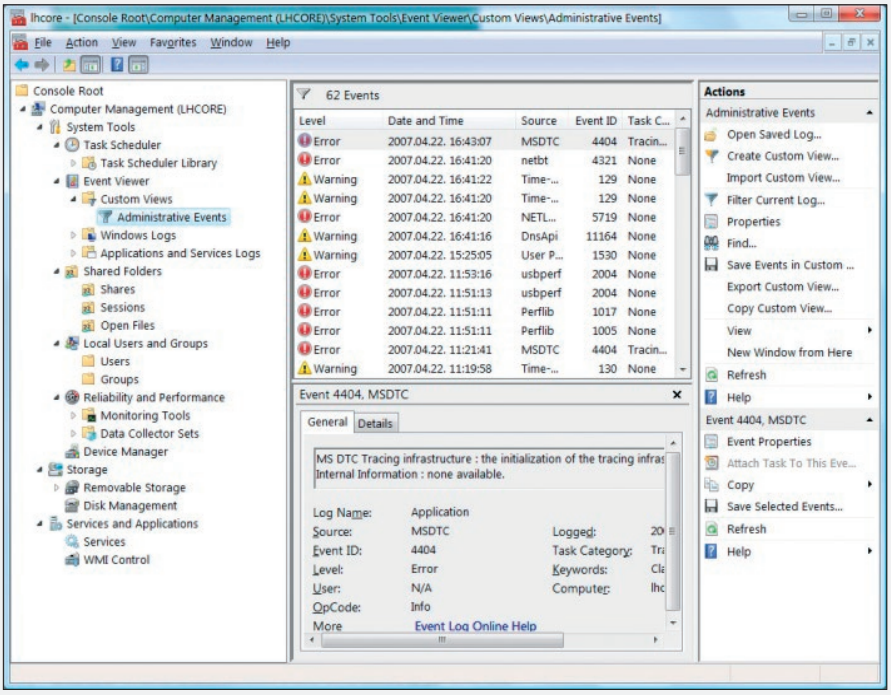
Ezután csont nélkül működik, ami azért is jó, mert például a vágólapon keresztül is letámadhatjuk a Server Core-t a megfelelő kötegelt vagy egyszeri parancsokkal. Egy másik lehetséges módszer az MMC-n keresztüli elérés, amely azonos tartományban, megfelelő jogosultsággal semmi extra tudást nem igényel.

A képről az is kiderül, hogy az újfajta, a Vistában már megismert Event Viewer-, Task Scheduler- vagy Performance Monitor-képességeket korlátozás nélkül használhatjuk a Server Core esetén is.

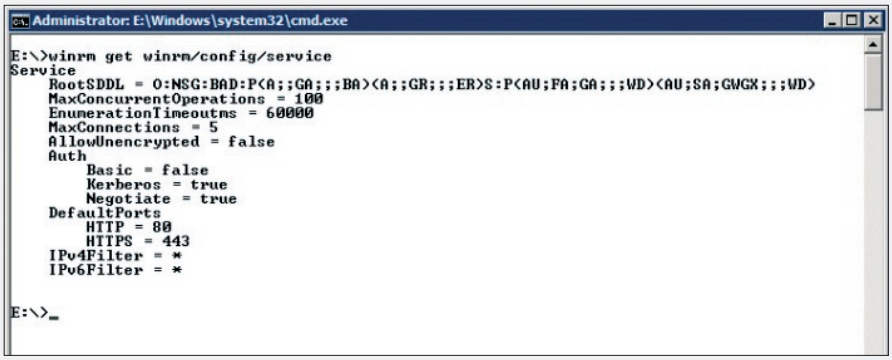
Ha viszont nem azonos tartományban vagyunk a kiszolgálóval, akkor elsőként szükség lesz erre a parancsra, ahhoz, hogy ne egy Access Denied sorozatba fussunk bele:

```
Net use * \\szerver_neve\c$ /u:user_neve
```

A harmadik módszerhez – a Windows Remote Management/Windows Remote Shell használatához – viszont célszerű azonos tartományban lenni a Server Core kiszolgálóval, hiszen ekkor könnyedén használhatjuk például a Kerberost a hitelesítésre, ami egyúttal az alapértelmezés is. De mi is igazából ez a páros? A Windows RM komponens része a Windows Hardware Management szolgáltatásnak, amellyel teljes körűen irányíthatjuk helyből vagy távolból a kiszolgálót. A szolgáltatás a WS-Management protokollt használja, hardveres diagnosztikát és ellenőrzést tesz lehetővé, és emellett a kiszolgáló szoftveres távvezérlésére is alkalmas a parancssorból. A csatlakozás tűzfalbarát módon (például HTTP vagy HTTPS), biztonságos körülmények között történhet meg, és persze többféle hitelesítési módszert (Basic, Digest, Kerberos) is alkalmazhatunk. Sokan úgy gondolják, hogy ez a megoldás csak a Vistákkal és a WS08-kiszolgálókkal működik, pedig nem, a Windows Server 2003 R2-ben is benne




*Így néz ki a Server Core Computer Management MMC-je egy Vistáról*




*A WinRM aktuális beállításai*

van ez a komponens, csak telepíteni kell. Más kérdés, hogy egy R2 WinRM listener (a szerveroldali „figyelő") beállítása igen bonyolult lett, de azért használható. A WS08-kiszolgálók és a Vista esetén viszont egyszerű az élet, a következő paranccsal indíthatjuk a szerveroldalon a szolgáltatás beállítását:

```
WinRM quickconfig
```

Ezzel a paranccsal elindítjuk és automatikus futtatásúvá tesszük a WinRM szolgáltatást, beállítjuk a HTTP listenert a WS-Management protokoll üzeneteinek fogadására és küldésére, valamint létrehozunk egy tűzfalkivétel szabályt (TCP 3190) a WinRM szolgáltatás részére. És ennyi! Ellenőrzés gyanánt győződjünk meg az alapértelmezett hitelesítés típusáról:

```
winrm get winrm/config/service
```

Példaként nézzünk meg néhány további parancsot. A Server Core rendszerpartíciója tartalmának listázásához a következő utasítást adjuk ki, mondjuk, egy Vista-kliensről:

```
winrs -r:http://szerver_neve dir c:\
```

A kiszolgáló újraindításához pedig gépeljük be ezt:

```
winrs -r:http:// szerver_neve shutdown -r
```

Visszatérve a felügyelet témakör elejére, meg kell említeni még egy-két helyben is használható eszközt is. Ide tartozik két Control Panel-elem, amelyek megmaradtak a Server Core-ban is.

1. Az idő/dátum beállítása: control timedate.cpl.
2. A területi beállítások: control intl.cpl

Ezeknél talán sokkal fontosabb viszont az az alkalmazás, amely egyaránt megtalálható minden Vistán és WS08 kiszolgálón is, azaz a Eseménynapló parancssoros változata, a WevtUtil, és amellyel láthatóan mindent el lehet érni, amit a GUI-s változattal.

## Szerepkörök, komponensek telepítése

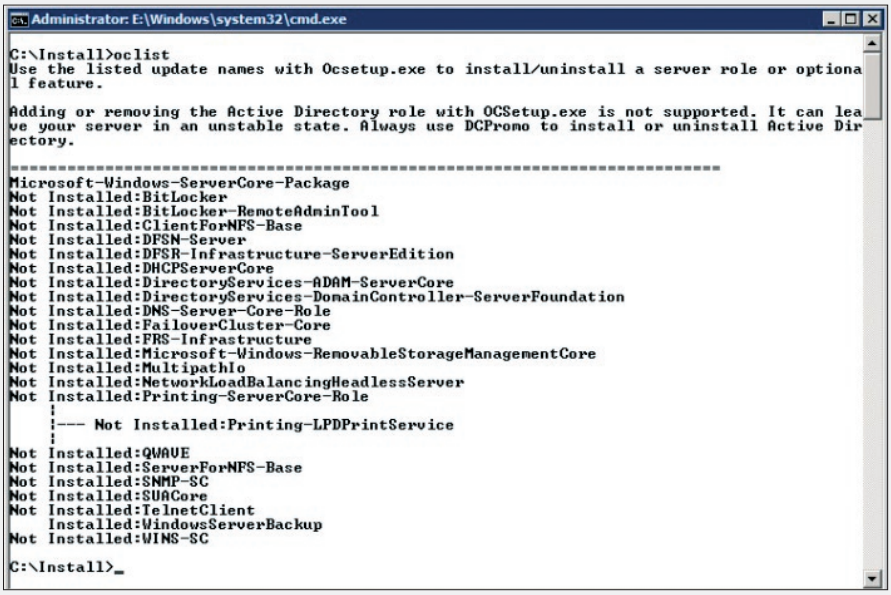
Fontos kérdés, hogyan tudunk alkalmazásokat és komponenseket telepíteni, illetve hogy melyek állnak rendelkezésünkre akár rögtön a telepítés után, azaz melyeket kell gyakorlatilag csak élesíteni? A fontosságuk szerint két részre szedett listát a cikk elején már láthattuk, most viszont az is kiderül, hogy a parancs gyakorlatilag ugyanaz mindkét csoportnál, azaz a

```
start /w Ocsetup
```

utasítás után a megfelelő szerepkör vagy komponens neve jön, a különbség maximum annyi, hogy a komolyabb szerepkörök nevei hosszabbak, például DNS-Server-Core-Role vagy File-Server-Core-Role és így tovább. Nagy segítségünkre lehet viszont az „Oclist" nevezetű parancs annak eldöntésére, hogy mi a szerepkörök pontos neve, illetve hogy melyeket telepítettük már fel. A következő képen szépen látszik, hogy ez egy friss Server Core, egyedül a mentési komponenst telepítettük fel.

A szerepkörök telepítésénél figyeljünk oda a gépelésre, mert az Ocsetup rendkívül érzékeny a kis- és nagybetűk közötti különbségre. Itt és most nem foglalkozunk tovább egy-egy szerepkör élesítés utáni konfigurálásával: a TechNet-blogon már esett szó (http://


*A WevtUtil parancsai és paraméterei*




*Az Oclist rendkívül hasznos parancs (a képen még nem a Beta 3 komponensei látszanak)*

www.microsoft.hu/technet) vagy a hivatalos Server Core-blogon (http://blogs.technet.com/server_core/) ezekről a lépésekről.

Ehhez a részhez még annyit kell hozzátenni, hogy a „nagy" Windows Server 2008-kiszolgáló Server Managere vagy egy GUI-s távoli telepítés/eltávolítás jelenleg nem áll rendelkezésre, és a híresztelések szerint nem is lesz ilyen lehetőség a végleges termékben sem.

Van viszont egy komoly elem, amely kívül esik az Ocsetup hatókörén, és egyéni törődést igényel. Az Active Directory telepítéséről van szó, amely nem túl egyszerű művelet, több előkészületre is szükség van hozzá. Először is, tényleg a fix IP kell, ezenkívül a séma preparálására is sort kell kerítenünk (változás nincs, maradt az adprep.exe és az ismerős kapcsolók a DVD-ről), úgyhogy csak óvatosan: ne feledjük, ez még mindig egy bétatermék, és a sémabővítés visszavonhatatlan folyamat!

Miután nem érhető el a grafikus felületű Dcpromo, muszáj az unattend módszert választani (amit egyébként a korábbi Windows-kiszolgálóknál is lehetett), de érdekes lenne megvizsgálni, mennyien éltek/élnek ezzel a lehetőséggel? Szóval össze kell kalapálnunk egy szövegfájlt, amely az ismert módon vezérelni fogja a telepítést, parancssori indítással. Egy példa egy szűkre szabott, de telepítésre tökéletesen alkalmas szövegfájlra:

```
[DCInstall]
AdministratorPassword =
AllowAnonymousAccess = No
AutoConfigDNS = Yes
CreateOrJoin = Join
CriticalReplicationOnly = Yes
DisableCancelForDnsInstall = Yes
DomainNetBiosName = xxx
RebootOnSuccess = Yes
RemoveApplicationPartitions = No
ReplicaDomainDNSName = xxx.yyy
ReplicaOrNewDomain = Replica
ReplicationSourceDC = zzz.xxx.yyy
SafeModeAdminPassword =
UserDomain = xxx.yyy
UserName = Administrator
Password =
```

Ezután már csak egy további teendőnk akad, az indító parancs kiadása:

```
Dcpromo /unattend:fájlneve.txt
```

A szövegfájlba felvehető paraméterekről a vonatkozó Windows Server 2003-dokumentumból tájékozódhatunk (http://tinyurl.com/2nbq95).

## Egyéb alkalmazások és a meghajtóprogramok

Nem sok egyéb alkalmazásunk van az eddig említetteken kívül, de ezek közül ami fontos, az például a Notepad (ami csak Beta 2-ben került bele, és csak a Beta 3-ban fog működni a Save/Save As) és az Open parancs rendesen, valamint például a Regedit.exe, ami szintén „újfiú", ugyanis eddig csak az import működött. Az Error Reporting (serverwer-optin.exe) szintén rendelkezésre áll, és van egy alkalmazásunk a biztonsági frissítések telepítéséhez (Wusa.exe), amely az .msu kiterjesztésű csomagokat kezeli.

Van viszont egy olyan apró, ám multifunkciós megoldásunk is – a SCRegEdit.wsf nevű szkript –, amely megtalálható a \Windows\System32-ben.

A segítségével engedélyezhető az AU-kliens, az RDP-elérés (korábban már szó volt erről), a távoli IPSec Monitor management, és például a DNS-rekordok prioritásszabályzásához is köze van. Valamint a /cli kapcsolója az összes, a Server Core-ban használható parancssori eszköz lehetőségeit kilistázza.

A korábbi példákban már láthattuk is ennek a scriptnek a használatát.

Végül legyen szó egy szintén kritikus területről, azaz a driverek telepítéséről, bár a tapasztalatok szerint a hardverek felismerésével és illesztésével abszolút nincs gond. De ha mégis, akkor a következő módszer szerint járjunk el:

1. Másoljuk be a meghajtóprogramot egy mappába
2. Pnputil -i -a mappa_neve\<driver>.inf
3. Újraindítás (nem mindig szükséges)

A jelenlegi meghajtóprogramok listázásához a következő régi ismerős parancsra lesz szükség (a szóköz a „driver" előtt szándékos):

```
sc query type= driver
```

A szolgáltatások vezérléséhez szintén az sc parancs a megoldás, például egy szerviz indítása történhet így is:

```
Sc config „RemoteRegistry" start= auto
```

Mint látható, a Windows Server Core mindenképpen érdemes lesz a figyelmünkre, hiszen biztonságosságának, egyszerűségének és alacsony gépigényének köszönhetően számtalan esetben lehet, hogy ezt fogjuk választani a teljes Windows Server 2008-változatok telepítése helyett.

*Gál Tamás*
(v-tagal@microsoft.com) Microsoft Magyarország

# Windows Server Virtualization

Egy közel száz kilobájtos kis réteg van készülőben – egy mikrokernel, amelyik képes az erőforrások (memória, processzor stb.) megosztására több operációs rendszer között. Mindezt a Windows Server 2008 egy szerepköreként kapjuk meg, gyakorlatilag teljesen ingyen. A szervervirtualizáció új generációja ez: a Windows Server Virtualization!

A virtualizáció már közel 30 éve jelen van a mainframe-rendszereken, azonban csak néhány éve jelentek meg az első virtualizációs technológiák az x86-os platformra. A mainframe-ek esetében az elsődleges cél a szoftverek visszafelé kompatibilitásának megőrzése volt, hogy a virtuális környezetekben akár évtizedekkel korábban elavult megoldások is futhassanak. Később egyre inkább teret nyertek a virtualizáció más irányú felhasználási módjai is, például az erőforrások egy fizikai gépen belüli elosztása a virtualizált operációs rendszerek között.

Az x86-os platformon is hasonló volt a helyzet – elsőként a desktop-virtualizációs megoldások jelentek meg, majd rohamosan fejlődni kezdett a szervervirtualizáció is. Majd – ahogy egyre többet tudtunk meg a virtualizáció lényegéről – a Terminal Services alapú megoldások is részben ide kerültek (megjelenítésvirtualizáció). Mára minden virtualizálható: a hálózat, a tárolórendszerek (például az iSCSI), de akár az alkalmazások is (például a SoftGrid).

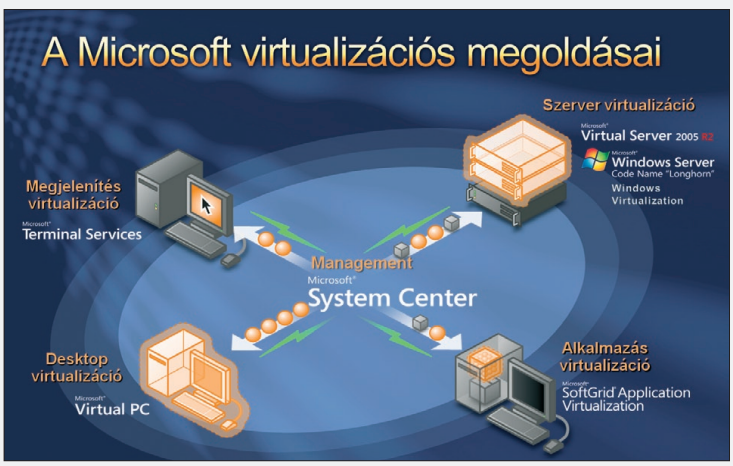
Nem meglepő ez a tendencia, hiszen egyre nagyobb az igény a rugalmasan változtatható informatikai rendszerek iránt. A virtualizáció talán legfontosabb célja ugyanis az, hogy rendszerünk összetevőit minél inkább elszigeteljük egymástól, és lehetővé váljon ezeknek az építőkockáknak a tetszés szerinti mozgatása, cseréje, frissítése.

## A szervervirtualizáció lehetőségei

Koncentráljunk most egy kicsit a szervervirtualizációra! Mire is jó ez nekünk? Milyen problémákra ad választ? Ha valaki még nem foglalkozott szervervirtualizációval, érdemes végiggondolnia az alábbi felhasználási lehetőségeket.

**Szerverkonszolidáció.** A szerverhardverek a legritkább esetben vannak folyamatosan kiterhelve a lehetőségeik határáig. Minden szolgáltatás máskor és eltérő mennyiségű számítási teljesítményt, illetve erőforrásokat igényel. Érdemes ezeket a különféle szolgáltatásokat minél kevesebb fizikai vasra központosítani, és azok skálázhatóságát és rendelkezésre állását biztosítani.

**A szolgáltatások folyamatos működésének biztosítása.** A cél itt igencsak egyszerű: szeretnénk minimalizálni mind a tervezett, mind a be nem tervezett rendszerleállások idejét. Minél kevesebbszer álljon le a rendszer, de ha le is áll, gyorsan helyre tudjuk azt állítani. Virtualizációval mindez könnyen megvalósítható, hiszen mind a fürtözésre, mind a virtuális lemezek és gépek replikáció-




*A Microsoft virtualizációs megoldásainak körképe*

jára és mozgatására is számtalan megoldás áll rendelkezésünkre, amihez egészen kényelmes rendszerfelügyeleti megoldások is elérhetők már.

**Dinamikus adatközpont.** Lehetőségünk van arra is, hogy az egy vasra konszolidált operációs rendszerek, illetve szolgáltatások között rugalmasan mozgathassuk az erőforrásokat, például a rendelkezésre álló memóriát, illetve a számítási kapacitást. Ha több szerverünk van, igény szerint másolhatjuk vagy pedig mozgathatjuk köztük a virtualizált gépeinket is.

**Fejlesztési és tesztkörnyezet.** Könnyen építhetünk olyan virtuális tesztkörnyezeteket, amelyeken kipróbálhatjuk az új szoftverváltozatokat: mennyire fognak helyt állni valós rendszerünkben. Ezeknek a virtuális környezeteknek nem kell külön fizikai szerverekre kerülniük – elférhetnek a már használatban lévő szervereken is, és mivel csak a teszt idejére van rájuk szükség, így erőforrásigényük is csak ideiglenes. A virtualizációnak köszönhetően tökéletesen izolálhatjuk a tesztrendszereket a valódiaktól (egy hardveren belül is!), de ha pont ennek az ellenkezőjére van szükségünk (például egy migráció tesztelésekor szeretnénk elérni az aktuális rendszert is), az is könnyen megvalósítható.

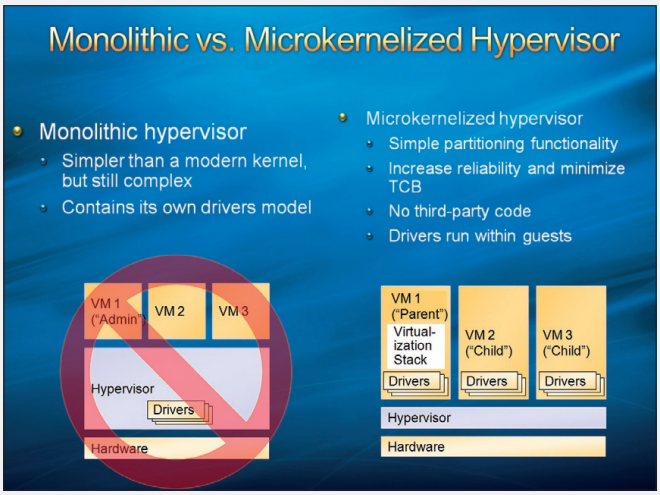
Sokan persze már csak mosolyogva legyintenek, olvasván ezeket a sorokat, hiszen már ismerik a ma elérhető megoldásokat, és használják is azokat, köztük a Microsoft Virtual Server 2005 R2-t, vagy például a VmWare megoldásait.

Nekik már sokkal gyakorlatiasabb problémáik vannak: a virtualizált rendszerek teljesítménye, az emulált és virtualizált hardverek használhatósága, a biztonság kérdése, a minél alaposabb izoláció és az egyszerű kezelhetőség és menedzselhetőség kerül előtérbe.

## Tervezési szempontok

A Windows Server Virtualization tervezésekor a korábbi céllal (a visszafelé kompatibilitás megvalósításával) szemben további, új szempontok is előtérbe kerültek.

Az első szempont az volt, hogy a rendszer a lehető legnagyobb biztonsággal működjék. Legyenek a különféle virtualizált rendszerek és a virtualizációt végző infrastruktúra egymástól teljesen elszigetelve, izolálva: ne érhessék el a virtualizált rendszerek egymás adatait, memóriáját; ne tudják egymás elől elvenni a rendelkezésre álló számítási kapacitást, hanem azt az infrastruktúra ossza meg köztük. A biztonság elérése érdekében az is fontos, hogy minél kisebb legyen a virtualizációs réteg kódja. Ez a réteg ugyanis mindenhez hozzáfér, és mindenhez van joga. A lehető legkisebbre kell csökkenteni a méretét, ezzel csökkentve egyúttal a támadási felületet is.




*A monolitikus és a mikrokernel alapú hypervisor közti különbségek*

A biztonság után legfontosabb szempontként a megbízhatóság állt: a virtualizációs réteg hibája vagy leállása ugyanis valamennyi azon futó virtuális gép leállásával jár együtt! Elég akár egy egyszerű rendszer-újraindításra gondolnunk. Virtualizáció nélkül egyetlen gép leállása csak egy adott szolgáltatás leállását eredményezi. Egy 20 virtuális gépet futtató vas leállása miatt azonban mind a 20 szolgáltatás azonnal leáll. Nagyon fontos tehát, hogy a rendszer minél megbízhatóbb legyen, másrészt legyen képes magas rendelkezésre állásra abban az esetben is, ha valamiért mégis leállás következik be. Többek között ezért is jár annyira kéz a kézben a virtualizáció és a fürtözés.

A fokozottabb megbízhatóság érdekében a Windows Server Virtualization az egyszerűségre törekszik. Ennek legfontosabb eszköze, hogy egyértelműen meghatározott, egymásra épülő rétegekre osztott a felépítése, és az ezek közti kommunikációs kapcsolatok száma alacsony, működésük a lehető legegyszerűbb. A hardverhez legközelebb eső rétegek (ezek rendelkeznek a legtöbb jogosultsággal) a lehető legkevesebb feladat elvégzésére képesek. Ezért maga a hypervisor egy nagyon apró mikrokernelként jött létre (ezzel később részletesen foglalkozunk), és kizárólag azokat a funkciókat tartalmazza, amelyekhez ténylegesen szükség van a legmagasabb jogosultságokra, illetve néhány olyan apró funkciót, amely az optimális teljesítmény eléréséhez teljességgel elengedhetetlen. Minden más a hypervisor fölött, a partíciókban fut – ezt virtualizációs réteg (virtualization stack) néven fogjuk a jövőben emlegetni.

Lényeges az eltérés például a VmWare ESX szerverhez képest, amely a minél nagyobb teljesítmény érdekében további drivereket és hardveremulációt is a hypervisor szintjére helyezett el – ez a monolitikus hypervisor-megközelítés. Ami azonban növeli a támadási felületet, növeli a leállások kockázatát (gondoljunk csak a hibás driverekre!), és gyakorlatilag teljes ellentétben áll a Microsoft által is képviselt minimalista, mikrokernel alapú hozzáállással szemben – mindezt néhány százaléknyi teljesítményért cserébe. A nyílt forrású hypervisor, a Xen is a Microsoft álláspontját osztja ebben a kérdésben, emiatt a két megoldás igen sok ponton képes lesz együttműködni – de erről még szintén lesz szó a későbbiekben.

A harmadik szempont a skálázhatóság volt – elérni, hogy a Windows Server Virtualization gyakorlatilag akármekkora gépen, tetszőleges méretű és számú virtuális gépet is képes legyen optimális teljesítménnyel kezelni.

## Követelmények és határok

A Windows Server Virtualization a következők meglétét igényli:

- x64-es WS08 Standard, Enterprise vagy Datacenter Edition, akár teljes, akár Core változatban a szülőpartícióra;
- x64-es processzor, Intel Virtualization vagy AMD Pacifica hardveres virtualizációtámogatással;
- hardveres DEP, azaz Data Execution Prevention (Intel XD/AMD NX).

És amire képes:

- 64 és 32 bites virtuális operációs rendszerek kiszolgálása (vegyesen is akár);
- akár 8 processzormag hozzárendelése bármely virtuális géphez;
- 2 terabájt memóriát oszthatunk szét a virtuális gépek között;
- tetszőleges számú virtuális gép futtatása (csak a hardverünk szab határt, nincsenek kódolt limitek).

Részletesebben az alábbi táblázatból tájékozódhatunk:

| | WS08 Standard x64 Edition | WS08 Enterprise x64 Edition | WS08 Datacenter x64 Edition |
|---|---|---|---|
| A támogatott fizikai processzorok száma | 1–4 processzor | 1–8 processzor | 1–64 processzor |
| A maximálisan támogatott memória | 32 gigabájt | 2 terabájt | 2 terabájt |
| Virtual Machine Live Migration | Nem | Igen | Igen |
| Cluster-támogatás | Nem | Igen | Igen |

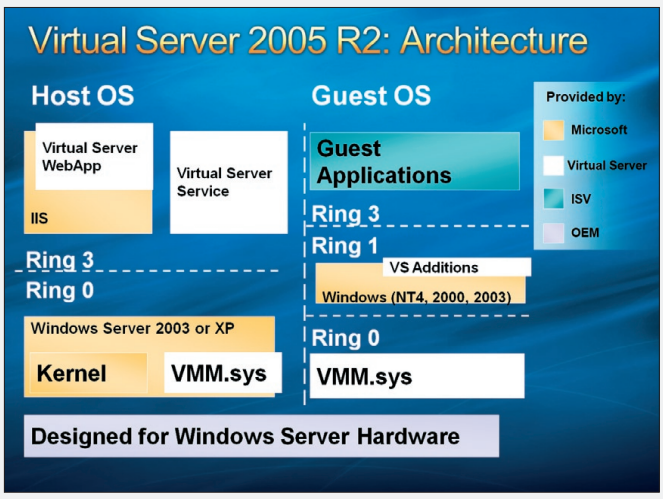
## Az architektúra

A Windows Server Virtualization egy teljesen 64 bites, mikrokerneles hypervisor alapú virtualizációs megoldás. A 64 bit talán egyből érthető is, bár rögtön két dolgot is jelent: egyrészt a virtualizációs réteg a 32 bites rendszerekkel szemben sokkal nagyobb memóriához fér hozzá, másrészt lehetőségünk van futtatni 32 és 64 bites virtuális operációs rendszereket, akár vegyesen is. No de mi az a hypervisor? Ennek megértéséhez először érdemes visszatekinteni egy kicsit a Virtual Server 2005-re.

Kétféle virtualizációt különböztetünk meg egymástól. A hagyományos (type 2 vagy host alapú) virtualizáció egy vékony réteget helyez el egy futó operációs rendszer (host) és a virtuális gépek (guest) között. Minden hardverrel kapcsolatos művelet keresztülhalad egyrészt a virtualizációs rétegen, majd magán a host-operációsrendszeren is, jelentős teljesítménycsökkenést eredményezve.

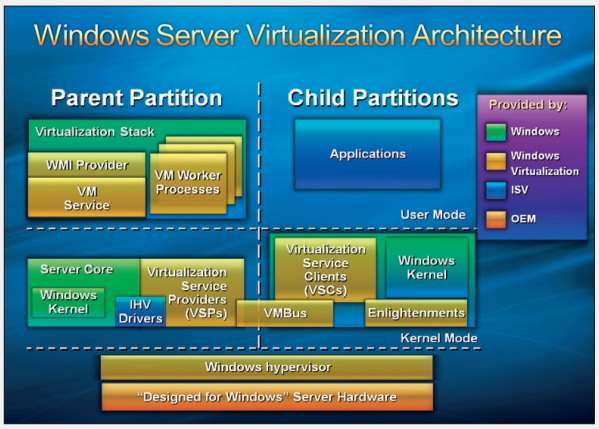
A modernebb, de még szintén erre a megoldásra épülő, úgynevezett hibrid virtualizációs technológiák esetében a virtualizációs réteg az operációs rendszerrel majdnem egy szinten található meg, és a hardverhívások többségét igyekszik minél közvetlenebb úton továbbítani a tényleges hardverekhez az operációs rendszer kihagyásával.

Erre jó példa a Virtual Server 2005 esetében a Virtual Machine Additions csomag, ami amellett, hogy átjárhatóvá teszi a virtuális guest és a host gépeket (egérkurzor-integráció, időszinkronizáció stb), a rendszer teljesítményén is javít azáltal, hogy még bootolás során egy driver segítségével átírja a rendszerhívások tábláját néhány (közel 6) ponton, hogy a leginkább hardverintenzív hívások teljesítménye virtualizált környezetben se lassuljon le számottevően.




*A Virtual Server 2005 R2 architektúrája*

A Virtual Server 2005 képes több mint ezer különféle operációs rendszer futtatására, méghozzá azok bármiféle módosítása nélkül. Ahhoz, hogy ez működjön, el kell hitetnie a virtualizációt végző rétegnek a virtuális operációs rendszerrel, hogy ő valóban teljes egészében kernel módban fut (ring 0), holott valójában a host-operációsrendszeren, Guest Kernel módban (ring 1). Ezt az emulációt nevezzük Ring Compressionnek, amit a kernel módban futó Virtual Machine Monitor (VMM) végez: a VMM figyeli a virtuális operációs rendszereket, és biztosítja, hogy azok ne csinálhassanak semmi butaságot (ne férhessenek a virtuális rendszerek hozzá más virtuális gépek, vagy akár a host gép memóriájához, adataihoz). Ez természetesen szintén csökkenti a rendszer teljesítményét, de hardveres virtualizációtámogatás nélkül más megoldás jelenleg nem létezik erre a problémára.



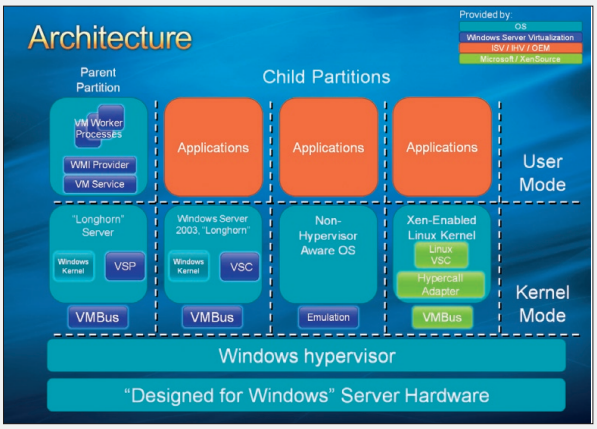

*A Windows Server Virtualization felépítése*

## A hypervisor

Ezzel szemben a hypervisor alapú megvalósítás (type 1 virtualizáció) esetében a virtualizált gépek és a hardver között semmi más nem áll, mint maga a hypervisor: egy vékony réteg, gyakorlatilag egy mikrokernel, ami közvetlenül a hardveren fut, nincs szüksége host-operációsrendszerre a működéshez. A Windows Server Virtualization felépítésére is ez jellemző. A hypervisor felel a virtualizált gépek futtatásáért, valamint azért, hogy számukra teljesen elszigetelt partíciókat alakítson ki az általunk beállított hardvereszközök, memóriaméret, számítási kapacitás, hálózati kártyák és egyéb beállítások alapján.

A Windows Server Virtualization kihasználja a hardveres virtualizációs megoldásokat, emiatt nincs többé szükség a Ring Compressionre. Miért is kellett a Ring Compression? Azért, mert a virtualizációs réteg és a virtualizált gépek nem futhatnak egy ringben

(biztonsági és izolációs okok miatt – hogy ne érhessék el egymás adatait közvetlenül), viszont az emulált operációs rendszereknek azt kellett hinniük, hogy ők valójában a 0-s ring-




*A szülő- és gyerekpartíciók viszonya*

ben futnak. Mi lenne az ideális megoldás? Ha a virtualizációt végző réteg a –1-es ringben futna. A hardveres virtualizáció pedig ezt teszi lehetővé: nincs szükség többé emulációra, és a teljesítmény sem romlik miatta.

## A szülőpartíció

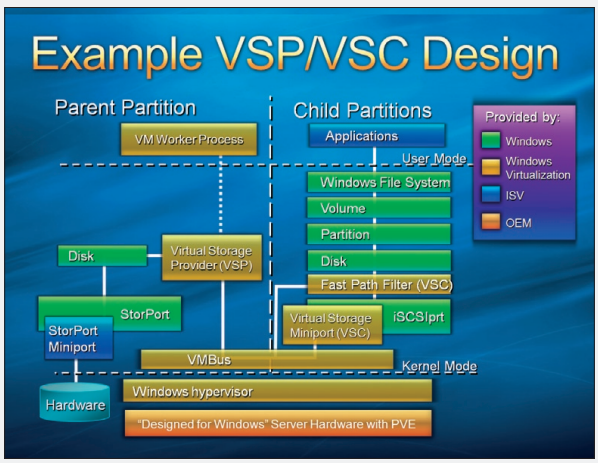
A Windows Server Virtualization esetében továbbra is van egy kiemelt jelentőségű virtuális gép – vagy más néven partíció –, de ennek a neve ezentúl szülő- (parent) partíció, és nem host. Ennek az az oka, hogy megváltozott a szerepe is. A szülőpartíció felelős valamennyi hardver és erőforrás kezeléséért, és ez végzi el a további partíciók létrehozásával, törlésével, felügyeletével kapcsolatos teendőket.

Gyakorlatilag a szülőpartíció amellett, hogy egy teljes értékű operációs rendszer, egyben a vékonyka hypervisor-réteg kiterjesztése is – itt található az a virtualizációs réteg, amit korábban már említettünk.

Miért előnyös ez? A driverek miatt! Ezzel a megoldással ugyanis nincs szükség speciális, virtualizációs driverek írására, hanem bármilyen driver, amelyik felmegy a szülőpartícióra, egyben elérhetővé válik a többi partíció (virtuális gép) számára is!

A szülőpartícióra kizárólag Windows Server 2008 telepíthető (Standard, Enterprise vagy Datacenter), de az akár Core változat is lehet. Ha a teljes Windows Server 2008-at telepítjük, akkor akár erről a partícióról közvetlenül menedzselhetjük valamennyi virtális gépünket egy MMC-s grafikus felületről, azonban ezzel csökkentjük a teljes rendszer teljesítményét és biztonságát. Ha viszont Windows Server Core-t telepítünk, akkor igaz ugyan, hogy a rendszer felügyelete csak távolról valósítható meg, de egy nagyon kicsi, erőforrásokat önmagában nem nagyon igénylő operációs rendszert kapunk, ami sokkal biztonságosabb, és valamennyi Windowsra írt drivert képes futtatni egyben. A Microsoft ajánlása az, hogy a szülőpartíció lehetőség szerint Core legyen.

Könnyen észrevehető, hogy csakúgy, mint a host gép esetén, a szülőpartíció is SPoF-ként (Single Point of Failure) viselkedik, vagyis ha az leáll, valamennyi virtuális gépünk is leáll egyben. Ennek kivédése érdekében érdemes fürtözni azt egy másik fizikai géppel, amelyen szintén egy Windows Server Core szülőpartíció található meg, valamint minden futtatott virtuális gép replikált változata is megtalálható rajta. Ez a megoldás gyakorlatilag analóg a Virtual Server 2005 R2 host clustering megoldásával. Viszont a fürtözés a Standard Windows Serverben nincs benne, ezért ezt a technikát csak Enterprise vagy Datacenter változatokkal használhatjuk.




*A hardvermegosztási alrendszer architektúrája*

A szülőpartíció teljesen ugyanúgy viselkedik, mint bármely más operációs rendszer. Ugyanúgy lehet patchelni is, akár Microsoft Update, WSUS vagy SMS segítségével.

## A hardvereszközök megosztása, emuláció

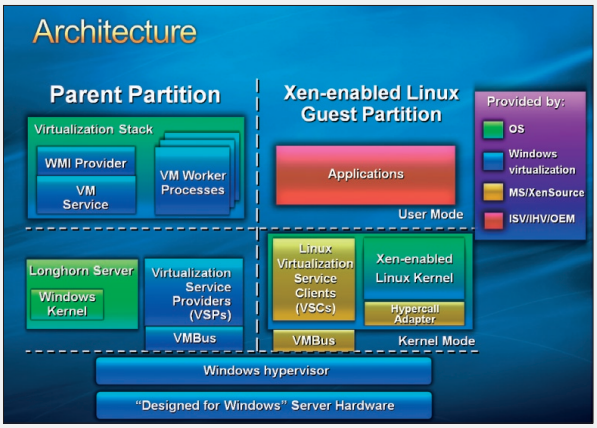
A hagyományos eszközemulációs megoldás nem éppen gyors, és nem is igazán skálázható nagyobb rendszerek esetén, különösen, ha például 20 virtuális gépünk fut párhuzamosan egy vason. A Windows Server Virtualization új hardvermegosztási architektúrája azonban választ ad erre is.

Mivel esélytelen elvárni bárkitől is, hogy emulációs drivereket fog készíteni régebbi hardverekhez a Windows Server Virtualizationhöz, ezért – mint korábban kifejtettük – a szülőpartícióra telepíthető drivereket használja az összes többi virtuális gép is. Az ehhez szükséges infrastruktúrához tartozik hozzá az alábbi három technológia.

**Virtualization Service Provider (VSP).** A szülőpartícióban fut – ez kommunikál a tényleges driverekkel, és osztja meg azt a virtuális gépekkel, multiplexerként működve. Például ha van egy fizikai hálókártyánk, amelyet 10 virtuális gép között szeretnénk megosztani, akkor a szülőpartíción található hálózati VSP elérhetővé fogja tenni azt a kártyát az olyan virtuális gépek számára, amelyeket beállítottunk, és mindegyik képes lesz egy időben használni azt. A Microsoft virtualizációs csapata már fejleszti azokat az általános hálózati, tárolóeszköz-, bemeneti és video-VSP-ket, amelyekkel tetszőleges eszközt tudunk megosztani egyszerűen driverük telepítésével a szülőpartícióra a többi virtuális gép között. A VSP-k telepítése automatikus a szülőpartícióra, amint engedélyezzük rajta a virtualizáció szerepkört.

**Virtualization Service Client (VSC).** Ezek a komponensek a gyerekpartíciókon futnak, és szintetikus eszközökként teszik elérhetővé azokat a hardvereket, amelyeket a szülőpartícióra telepítettünk, és megosztottunk az adott gyerekpartícióval. Minden gyerekpartíción megtalálhatóak ezek a VSC-komponensek, annak megfelelően, hogy milyen VSP-ket szeretnénk használni rajtuk (párban vannak). A VSC-k telepítése nem automatikus, az Integration Components telepítésével együtt kerülnek fel a virtuális gépünkre.

Szintetikus eszközön azt értjük, hogy egy hálókártya nem „DEC/Intel Ethernet Card”-ként, hanem „Microsoft Virtual Network Adapter”-ként jelenik meg. Ez azon kívül, hogy egy általánosabb név, azt is jelenti, hogy nem valóban létező hardvereszköz képességeit emulálja a VSC–VSP páros, hanem lehetőség van arra, hogy egy fizikai eszköz lehetőségeit messze túlszárnyaljuk ezzel a megoldással, akár új képességeket fejlesszünk ki hozzá. (Erre láthatunk egy példát a tárolóeszközöknél.)




*Xen alapú Linuxok és a Windows Server Virtualization együttműködése*

**VMBus.** Egy olyan, memórián keresztül működő, nagyteljesítményű sínrendszer, amelyik a partíciók közötti adatkommunikációért felelős. Ezen keresztül kommunikálnak egymással a VSC-k, illetve a VSP-k, de a hypervisor maga nem érhető el ezen keresztül. A VMBus nem emulált hardverként viselkedik, és nem is jelenik meg a szintetikus eszközök sínrendszereként a hardverek között az eszközkezelőben.

Ezek a megoldások nagyban növelik a virtualizált rendszerek teljesítményét, különösen az IO alrendszerrel kapcsolatos műveletek esetén, és lehetővé teszik olyan eszközök megosztását és virtualizálását is, amelyekre korábban nem volt mód. Mégis, joggal merülhet fel a kérdés: ezek szerint minden eszközemuláció megszűnt? Nincs rá többé szükség?

A válasz: nem. Továbbra is szükség van hardveremulációra. Mivel egyetlen operációs rendszer sem tartalmazza alapból a VSC-komponenseket (még a Windows Server 2008 sem!), ezért legalább a virtuális gépek telepítésének idejére szükség van hardverek emulálására. Emiatt továbbra is ezernél több marad a támogatott és telepíthető operációs rendszerek száma Windows Server Virtualization alapokon is. Hasonló módon a bootolás korai szakaszában is szükség van az emulált eszközökre, hiszen a VSC-k csak egy kicsivel később töltődnek be és aktiválódnak. Amint a VSC-k betöltődnek, teljesen átveszik az irányítást az emulált eszközöktől.

## Mi a helyzet a Linuxokkal?

Mivel rengetegen kérik, hogy a Microsoft virtualizációs megoldásai ugyanolyan jól támogassák a Linux operációs rendszereket, mint a Windowsokat, ezért nem lenne megfelelő megoldás, ha Linux alatt csak emulált eszközök lennének elérhetőek. A XenSource ezért a Microsofttal kötött partneri megállapodásának értelmében elkészíti a VSC-k linuxos változatait a legelterjedtebb Linux-disztribúciókra is (egyelőre Novell Suse és Red Hat), ezáltal Linuxokon is elérhetőek lesznek a nagysebességű szintetikus eszközök a VSP-ken és a VMBuson keresztül.

Ráadásul, mivel a XenSource által készített, szintén mikrokernel és hypervisor alapú virtualizációs megoldás nagyon hasonló felépítésében és koncepciójában a Microsoft-féle Windows Server Virtualizationhöz, és mindkét cég megoldásai használják a VHD fájlformátumot a virtuális gépek lemezeihez, ezért a Xen és a Windows Server Virtualization között a virtuális gépek cseréje meglehetősen egyszerűnek ígérkezik.

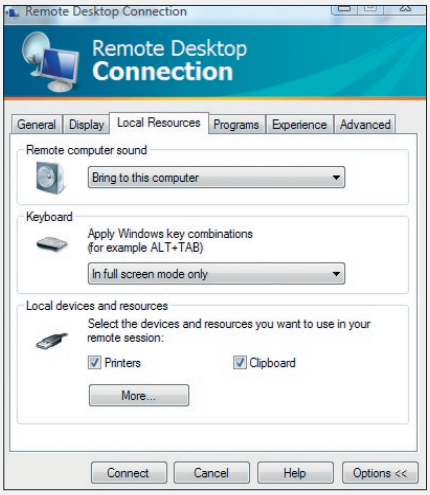
## USB, hang, videó és a BIOS

Érdekes kérdés, hogy vajon mi a helyzet az olyan egzotikumokkal, mint például az USB-eszközök, a hangkártyák vagy a 3D grafikus kártyák. Nézzük őket szépen sorjában!

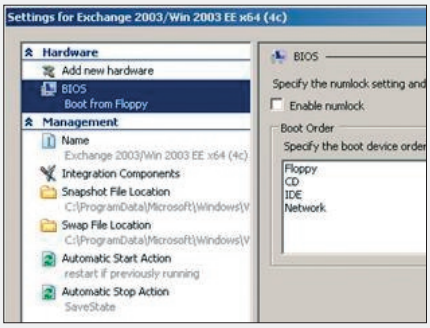
Egyelőre a virtualizációs csapat nem fejezte be a teljeskörű USB-támogatást, így az a Windows Server Virtualization első változatában nem lesz elérhető. Azonban mivel a virtuális gépeket mostantól RDP-n keresztül lehet elérni (a VMRC a továbbiakban már nem opció), lehetőségünk van akár smartcardok, akár USB-s tárolóeszközök használatára is a hagyományos RDP-kapcsolat beállításain keresztül.

Hasonló a helyzet a hangkártya esetén – bár a szervereken ritkán van szükség hangkártyára, és a Windows Server Virtualization nem is emulál jelenleg hangkártyát a virtuális gépeken, az RDP képes emulálni a hangkártyát a kapcsolat idejére.

A grafikus kártya kérdése sem teljesen egyértelmű – mert nem szokás ugyan szervereken 3D-grafikát használni, és általában egyszerű, 2D-kártyákat találunk a szerverekben, mégis




*Hangkártya és más eszközök emulálása RDP-n keresztül*




*A BIOS összes beállítása elérhető az MMC-ről*


*Tárolóeszközök és smartcardok megosztása RDP-n keresztül*

szükségünk lehet például egy Aero felülettel rendelkező Windows Vista virtualizálására és megfelelő megjelenítésére is. Maga a Windows Server Virtualization ugyanazt az S3 Trio kártyát emulálja, mint a Virtual Server, azonban ha egy Aero-képes Windows Vistáról RDP-zünk rá egy virtuális Vistára, akkor fogjuk tudni használni a virtuális gépen is az Aerót.

Azáltal, hogy a VMRC protokollt nem használjuk a továbbiakban, felmerül még pár apróság: például hogyan érjük el a virtuális gépek BIOS-át? A válasz: sehogy. Nincs többé mód a BIOS hagyományos elérésére, viszont helyette minden beállítás elérhető a Windows Server Virtualization MMC-jén keresztül. Ugyanilyen módon tudjuk beállítani a bootolandó eszközök listáját, sorrendjét is, és bootolhatunk akár lokális lemezről, USB-, firewire-, SAN- és NAS-eszközökről is.

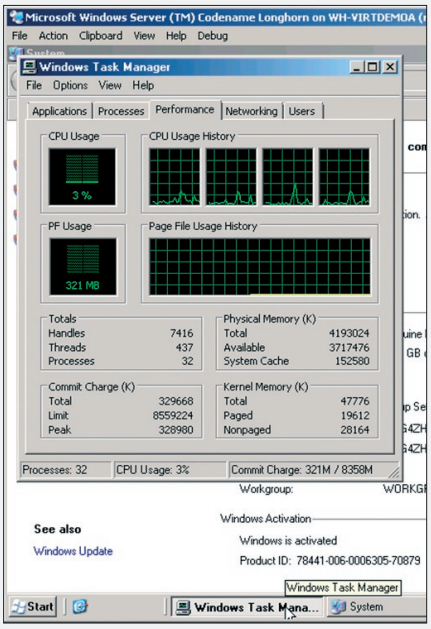
## Menet közbeni bővítés

A Windows Server Virtualization alatt futó virtuális gépekhez futás közben allokálhatunk további memóriát, processzormagokat, tárolóeszközöket, illetve hálózati kártyákat is. Ehhez azonban ezt a gyerekpartíción futó operációs rendszernek is támogatnia kell. A szülőpartíció, mivel csak Windows Server 2008 lehet, nem okozhat problémát, az mindegyik bővítési módszert támogatja. A XenSource megállapodásnak köszönhetően a virtualizált Linuxok is képesek lesznek a menet közbeni bővítés használatára, de ez kernelverziónként és disztribúciónként változó.

Az eszközök menet közbeni eltávolítására is van lehetőség hálózati csatolók és tárolóeszközök esetében, azonban a processzormagok és a memória eltávolítását a Windows Serverek jelenleg nem támogatják – de ennek megoldására is léteznek kerülőutak.

## Processzorok, memóriakezelés

A Windows Server Virtualization 1, 2, 4, illetve 8 processzormag hozzárendelését támogatja egy adott virtuális géphez. A virtuális gép nemcsak azt látja, hogy hány magot adtunk neki, azzal is pontosan tisztában van, hogy az hány fizikai processzorhoz tartozik. Erre feltétlenül szükség volt, hiszen a licencelési kérdések esetében sokkal kedvezőbben járunk így egyes gyártókkal, például a Microsofttal is, amelyek továbbra is processzorszám és nem processzormagszám alapján licencelik termékeiket.




*Egy négymagos virtuális gép Windows Server Virtualization alatt*

Azon túl, hogy a Windows Server Virtualization szinte korlátlan mennyiségű memóriát képes használni, megjelent néhány újabb képesség is a rendszer részeként.

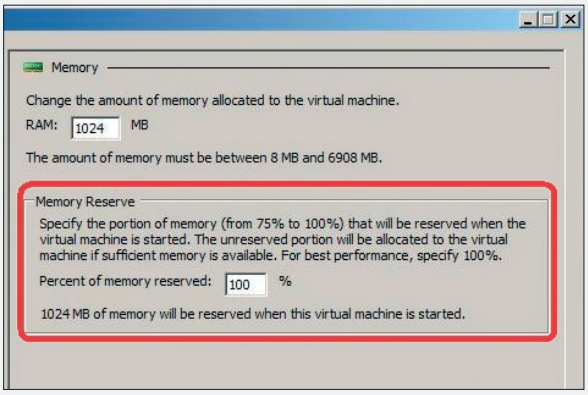
Az egyik ilyen újdonság a Page Sharing technológia, amelynek révén a virtuális gépek között lehetővé válik az azonos memórialapok megosztása. Ez azonos operációs rendszerek esetén rendkívül sokat segíthet, hiszen ugyanazt a kernelt és nagyjából ugyanazokat a rendszerszolgáltatásokat használják mind. Természetesen a Page Sharing csak a teljesen megegyező memórialapokat osztja meg a gépek között, ha valamelyik gép picit is eltér a többitől, akkor az az eltérés csak a saját memóriatartományában lesz elérhető. A megoldás előnye, hogy kevesebb memóriára lesz szükségünk, ha sok hasonló virtuális gépünk van. Hátránya: minimális teljesítménycsökkenéssel számolhatunk.

A másik újdonság a Memory Reserves funkció: lehetőségünk van arra, hogy a virtuális géphez rendelt memória egy adott százalékát ne adjuk oda azonnal a virtuális gépnek, csak akkor, ha tényleg szüksége van rá, és van még szabad fizikai memóriánk. Ha már nincs, akkor a Windows Server Virtualization virtuális memóriát fog létrehozni az adott virtuális gép számára.

Ennek a két funkciónak akkor van igazán értelme, ha tesztrendszereket szeretnénk virtuálisan kiépíteni, és nem rendelkezünk korlátlan mennyiségű memóriával. Éles környezetben azonban mindkettő jelentősen rontható a teljesítményen, ezért ezeket a funkciókat ne használjuk akkor, ha a lehető legjobb teljesítményt szeretnénk elérni. Éppen ezért a két beállítás kéz a kézben jár:

- Ha a legjobb teljesítményt szeretnénk, állítsuk 100 százalékra a Memory Reserves opciót. Ekkor a virtuális gép azonnal és fixen megkapja az összes számára szükséges memóriát. Ilyen esetben a Page Sharing is ki van kapcsolva, hiszen a teljesítményre optimalizálunk.
- Ha szeretnénk használni a Page Sharinget is, és inkább több memóriára van szükségünk, akár a teljesítmény kárára is, állítsuk a Memory Reserves opciót 100 százaléknál kevesebbre. A minimum, amit beállíthatunk 75 százalék. Ilyen esetben a virtuális gép azonnal megkapja a számára beállított memória 75 százalékát, majd ha azt felhasználta, kaphat még a fennmaradó fizikai memóriából.

Rosszabb esetben virtuális memóriát fog kapni, ha már nincs több szabad memória, amihez a virtuális gép hozzáférhetne.




*A Memory Reserve beállítási lehetőségei*

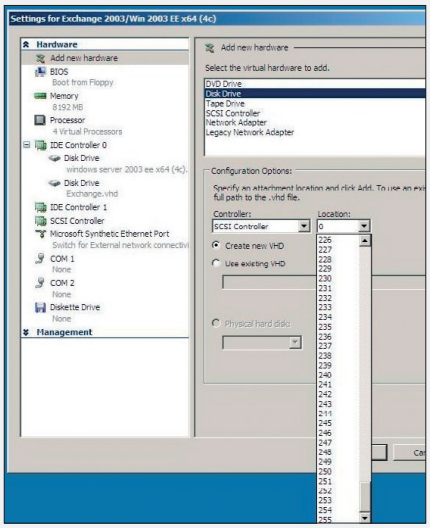
## Tárolóeszközök

A VSP/VSC architektúrának köszönhetően a tárolóeszközökkel kapcsolatban számtalan olyan újdonság érkezik, amely kihasználja a szintetikus eszközök lehetőségeit, és új képességekkel jelentkezik a korábbi, technológiailag limitált driverekhez képest.

Korábban Virtual PC és Virtual Server alatt az emu-

lált IDE-vezérlő legfeljebb 127 gigabájtos merevlemezeket volt képes kezelni. Ez a határ az új szintetikus eszközzel 2 terabájt, csakúgy, mint az SCSI-eszközök esetében. Ezenkívül most már ugyanolyan gyors a szintetikus IDE-vezérlő is, mint az SCSI-vezérlő (az emulált viszont még mindig lassabb – érdemes telepíteni a VSC-ket mielőbb!).

Az SCSI-vezérlő is fejlődött, most már vezérlőnként 256 virtuális merevlemezt használhatunk egyszerre, és ezek egyenként 2 terabájtos méretűek is lehetnek. Linux, Windows Server 2003 és Windows Server 2008 alatt legalább 2 SCSI-vezérlőt lehet majd használni (a guest clustering támogatása miatt). Azokon a rendszereken, ahol nem érhető el szintetikus SCSI-vezérlő, jelenleg 4 IDE-eszközt lehet legfeljebb használni, de az új korlátokkal ez akár 8 terabájt tárhelyet is jelenthet.




*Akár 256 eszközt is köthetünk a virtuális SCSI-vezérlőre*

## Hálózatkezelés

Ezentúl virtuális gépenként 8 hálózati csatolót lehet majd használni, de ehhez szükség van a megfelelő VSC-k telepítésére, ugyanis ez a határ csak a szintetikus eszközök esetén érhető el. Emulált eszközökkel továbbra is 4 hálózati kártya a maximum.

A Virtual Serverben már volt lehetőség arra, hogy virtuális hálózatokat definiáljunk és kössünk hozzá virtuális gépeink hálózati csatolóihoz.

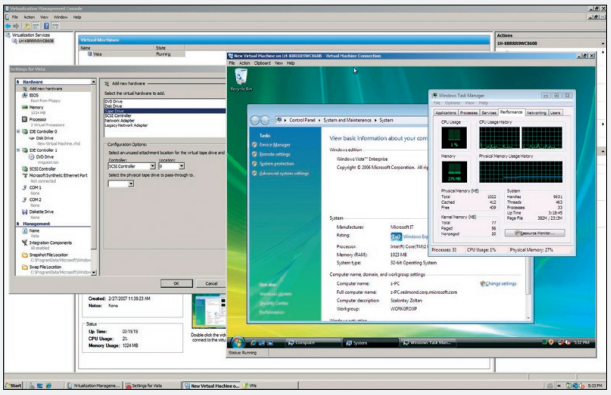
Ez a Virtual Server esetében nem volt más, mint egy uplinkkel rendelkező egyszerű hálózati hub. Ami azonban azt is jelenti, hogy az azonos (virtuális) hubra kapcsolódó gépek képesek az egymásnak küldött adatokba belehallgatni, és ez nem éppen biztonságos megoldás. A Windows Server Virtualization esetén már nem hub, hanem egy virtuális switch áll rendelkezésünkre, és ez már csak azokra a portokra küldi el a csomagokat, ahova tényleg szükséges, és nem mindre, mint egy hub.

Szintén újdonság, hogy már lehetőség van VLAN-ok használatára is, valamint akár a NAP-pal is képesek együttműködni a virtuális hálózatok, igaz, még csak IPSec alapokon.

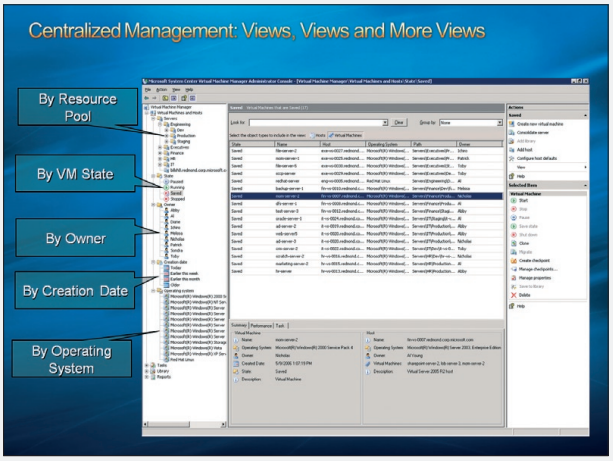
## Felügyeleti újdonságok, távoli elérés

Történt jó néhány változás a virtuális gépek felügyeletével kapcsolatban is. Az első, hogy a webes adminisztrációs felület helyett egy MMC fogad minket, ami természetesen távoli gépről is tökéletesen elérhető.

Hasonlóképpen újdonság, hogy a VMRC prorokollt is lecserélték, és helyette minden az RDP-vel, a Remote Desktoppal érhető el. Ennek számtalan előnye van a VMRC-vel szemben. Lehetőségünk lesz olyan virtualizált operációs rendszerre is csatlakozni Remote Desktoppal, ami amúgy nem támogatja a Terminal Servicest, és ugyanúgy elérhető lesz egy ActiveX Control az RDP-hez, mint ahogy a VMRC esetében is megszokhattuk.



*Akcióban a Windows Server Virtualization*




*A System Center Virtual Machine Manager kényelmesebbé teszi a virtuális gépparkok kezelését*

A Windows Server Virtualization valamenynyi összetevője WMI segítségével lesz scriptelhető, ha pedig mélyebbre szeretnénk ásni a rendszer lelki világába, böngésszük át a HyperCall API-kat.

Ami még a felügyelet kapcsán érdekes lehet: csakúgy, mint a Virtual Server 2005 R2 SP1, a Windows Server Virtualization is támogatja már a Volume Shadow Copy szolgáltatását, így lehetőség van a VHD-k futás közben történő mentésére is (Volume Snapshot), valamint vannak eszközeink a VHD-állományok megnyitására is (de csak ha éppen nem használjuk őket), hogy abban kézzel végezzünk el módosításokat.

Ha pedig egy teljes virtuális gépparkot szeretnénk megfelelően felügyelni, szükségünk lesz a System Center Virtual Machine Managerre is, ami mind Virtual Server alapú, mind pedig Windows Server Virtualization alapú gépek felügyeletére is alkalmas.

## Zárszó

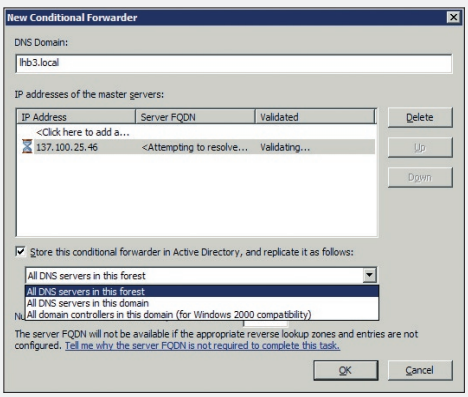
Érdekes technológia lesz a Windows Server Virtualization, az már biztosan látszik – az első publikus bétaverzió 2007 második felében érkezik, a végleges változat pedig a Windows Server 2008 után legfejlebb 180 nappal lesz elérhető. Aki szeretne élőben is megismerkedni a rendszerrel, látogasson el a http://tinyurl.com/yss3e2 URL-re.

*Budai Péter*
(i-pbudai@microsoft.com) Microsoft Magyarország

# Összefűzve – új hálózati technikák

A következő oldalakon egy jól megkevert összeállítást nyújtunk, amelynek elemei azonban két ponton mégis összekapcsolódnak. Az egyik pont a Windows Server 2008, a másik pedig a hálózat – akár névfeloldásról, biztonságos távoli elérésről, az alkalmazások sávszélesség-szabályozásáról, akár pedig a terminálszolgáltatás számos újdonságáról van szó.

Kezdjük a DNS-sel, ahol két szerveroldali változást kell megemlítenünk. Elsőként a zónák háttérben történő betöltése tekintendő fontos előrelépésnek, ami talán eldönti „...a fájlban vagy az AD-ban tartsuk a DNS-zónákat” című vitát is – tudniillik az előbbi esetében ez az új lehetőség nem áll rendelkezésre. Ha viszont a címtárban tartjuk a zónáinkat, akkor a Windows Server 2008 képes lesz ezeket egy DNS-restart után a háttérben, szeparált szakaszokban, aszinkron módon betölteni. Így az eddigiekkel szemben képes lesz a folyamat közben is válaszolni a beérkező névfeloldási kérésekre, azaz nem bicsaklik bele egy esetlegesen nagyméretű zóna egyszerre, lassan történő betöltésébe. Sőt, ha olyan kérés/regisztráció érkezik, amely egy olyan zónára vonatkozik, ami még nincs a memóriában, akkor a kliens óhaja magasabb prioritást kap és a DNS-szerver azonnal kiszolgálja a kérést. De újra hangsúlyozni kell, hogy ez csak a címtárban tartott DNS-zónákra igaz, a fájlban tartott zónáknál marad a szekvenciális feldolgozás.




*Feltételes továbbítók beállítása*

Egy másik szerveroldali újdonság a feltételes továbbítók újszerű használata. A DNS MMC-ben, a faszerkezetben a zónatípusok között egy új mappát láthatunk, „Conditional Forwarders” néven. Itt kell felvennünk a különböző továbbítókat (a megszokott helyen is lehet persze), és a felvétel után ezeket a jobb oldali keretben megjelenő listában rögtön láthatjuk is. Ennél talán fontosabb viszont az, hogy integrált AD-zóna esetén lehetőségünk van a címtárban tárolni és ebből következően replikálni is a továbbítókkal kapcsolatos adatokat. A replikáció többféle felállásban is működhet, azaz például minden tartományvezérlő DNS-szerverre (amely legalább Windows Server 2003) vagy éppen minden DC-re a tartományban.

## Secure Socket Tunneling Protocol (SSTP)

Sokan lesznek, akik azt mondják majd miután beüzemelték ezt a szolgáltatást: „Na végre!” Ezzel nem a művelet hosszúságára, hanem természetesen e komponens szükségességére gondolunk. Mert az SSTP különleges megoldás, és jól beleillik abba a tendenciába, amely alapján – ez egy személyes vélemény – pár generáció (platform) múlva csak a HTTP és a HTTPS protokollokat kell majd kinyitni a tűzfalakban. Leplezzük le végre: a hagyományos VPN-kapcsolatok helyett/mellett a Windows Server 2008-ban, HTTPS-en keresztül is képesek leszünk teljes értékű VPN-kapcsolatokat kezdeményezni. A jövő tényleg a HTTP/HTTPS alapú kommunikációé: erre bizonyíték az is, hogy még ebben a cikkben meg fogunk említeni egy másik területet is, ahol szintén van már passzoló megoldás, de sokan ismerjük és favorizáljuk a többi (... over HTTPS/HTTPS) megoldást is.

Legalább három fő érvet fel tudunk sorakoztatni a VPN over HTTPS (az SSL VPN elnevezés a hivatalos) mellett és a hagyományos típusokkal szemben:

1. A speciális VPN-portokat nem tudjuk, nem lehetséges minden körülmények között használni, egyszerűen egy sereg helyen (pél-

dául szállodákban, publikus helyeken vagy más cégek hálózatában) tiltják.

2. Bármelyik hagyományos VPN-típust nézzük, a tunnelt a legtöbb esetben „át kell vezetni" egy NAT-szerveren. Ez olykor kisebb, olykor nagyobb (L2TP) problémát is okozhat.

3. A VPN-kapcsolatok tipikusan „egy végpont, egy csomópont" típusúak. Ha a két helyszín LAN IP-tartománya megegyezik, és közöttük NAT-ot alkalmazunk, akkor szintén konfliktus van.

Persze, az utóbbi két problémára léteznek ajánlott és működő megoldások, de könnyen belátható, hogy egyetlen portot kinyitva, a NAT és más hálózati nehézségek nélkül egyszerűbb lenne működtetni egy VPN-infrastruktúrát.

Az SSTP viszont egy, az alkalmazási rétegben működő protokoll, tipikusan két program közötti kommunikációra felkészítve – ugyanakkor egy hálózati kapcsolaton belül akár többre is (gyakorlatilag a teljes hálózatban), ergo jobban képes kihasználni a sávszélességet. Az SSTP ugyanarra az SSL-háttérre támaszkodik, mint például az L2TP/IPSec (egymás mellett mindhárom típus jól elfér), és ugyanúgy a TCP 443-as portot használja. De tudnunk kell, hogy úgy, ahogy az L2TP IPSec nélkül, az SSTP SSL nélkül sem más, mint egy kicsit különlegesebb tunneling-protokoll. Hátrányai közé tartozik még az is, hogy a Site-to-Site kapcsolatokban nem vehetjük majd hasznát.

A lehetséges hátrányok után lássuk az előnyöket:

- Nincs szükség külön kliensre, és nincs szükség például extra-IP címekre.
- Mivel böngészőből megy a kapcsolat, csak TCP/IP-re lesz szükség, és ellentétben a sima SSL-lel, itt a teljes session esetén működik a titkosítás.
- Teljesen transzparens a felhasználó számára, és nem kell speciális útválasztást, illetve metrikát sem használnunk. Nem számítanak akadálynak a kapcsolat két pontja között működő routerek, tűzfalak, webproxyk és NAT-szerverek. Nem függ a kapcsolat olyan extra protokolloktól, mint a PPTP GRE vagy az L2TP ESP.
- Kompatibilis az IPv6-tal, a NAP-pal, az RRAS-sal, akár a multifaktoros azonosítással is.
- Az alkalmazási rétegbeli működés miatt majd igazán kényelmesen szűrhetjük a forgalmat egy olyan tűzfallal, amely erre képes (például ISA Server 2004/2006), persze ehhez a speciális SSL Bridging módszerrel kell majd publikálnunk.

Az SSTP-t tehát a Windows Server 2008 már tartalmazza, de alapértelmezés szerint nincs élesítve. Igazából nem kell semmi extrára gondolnunk, az RRAS-ban a szokásos módon összehozunk egy VPN-szervert, egy kattintással beizzítható az SSTP a Network Policy Serverben (NPS).

Ami viszont fontos: még az RRAS indítása előtt rendelkeznünk kell megfelelő kiszolgáló tanúsítvánnyal, hiszen az SSTP HTTPS listenerjéhez ez alapfeltétel.

Érdemes kiemelni, hogy a napokban megjelent Windows Server 2008 Beta 3 már működtethető SSTP-kiszolgálóként, de az első kliens csak később érkezik meg hozzá, ami nem más, mint a Vista SP1 (amely a WS08 RTM-mel együtt a 2007-es év vége felé fog megjelenni).

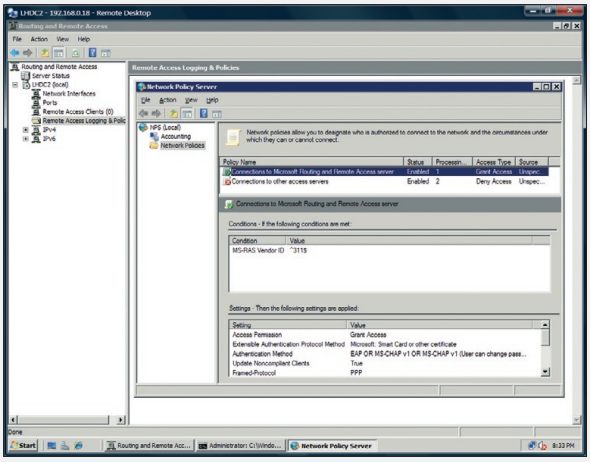
## A QoS szolgáltatás

A házirend alapú Quality of Service (QoS) komponens alapértelmezés szerint része a Windows Server 2008-nak. Segítségével egyszerűen megoldható a WS08- vagy Vista-kliensek esetén az alkalmazások sávszélesség-szabályozása.

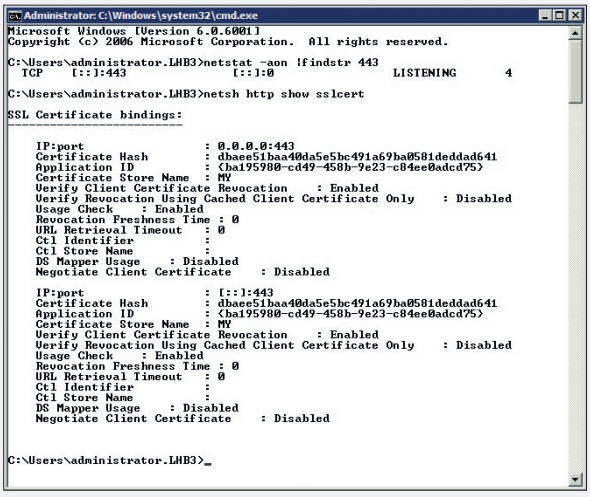
Az összes beállítás és e beállítások terjesztése is a Csoportházirend segítségével történik. Ennek a módszernek számtalan előnye van, a központi üzemeltetéstől kezdve egészen a megfelelő gépek, gépcsoportok, telephelyek, tartományok, felhasználók és csoportjaik kiválasztásáig. Külön előnynek számít, hogy mivel a korlátozás az alkalmazási rétegben történik, a meglévő alkalmazásokat semmilyen módon nem kell megváltoztatni vagy frissíteni ahhoz, hogy ezt a technológiát igénybe vehessük.

Egy példán keresztül szeretnénk bemutatni a QoS-házirendek alkalmazhatóságát. Képzeljük el, hogy egy telephelyes környezetben a telephelyi kliensek rendszeresen, időzítve mentik a rendszerállapotot a központi szerverre, ezzel szépen „eldugítják" a WAN-kapcsolatot, amely aztán rendszeresen fennakadásokat okoz az egyéb hálózati forgalomban, ezért aztán szükségesé válik egy QoS-házirend bevezetése.

Szerencsére a közbülső hálózati eszközök ismerik a DSCP-t (RFC 2474), azaz a prioritásszabályozást konfigurálhatjuk ezeken az eszközökön is. Ez azért fontos, mert ha el-



*Az Network Policy Serverben kell engedélyeznünk az SSTP-t*

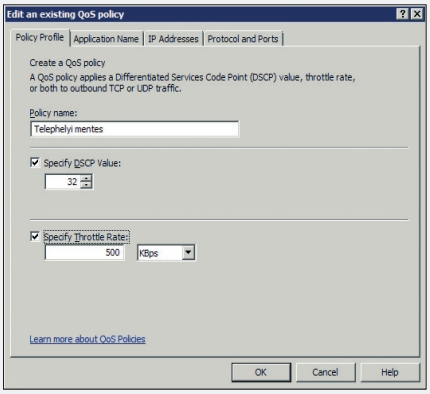



*A portfoglalás, illetve a tanúsítvány ellenőrzésének eredménye*

indítjuk a GPMC-n belül az adott GPO-ban a QoS-varázslót (Computer Configuration/Windows Settings/Policy based QoS), akkor a legelső panelen rögtön láthatjuk, hogy a konkrét sávszélességértéken kívül a DSCP-értéket is szabályozhatjuk. De miért is? Nos, mivel esetleg számtalan QoS-házirendünk is

lehet, ezek között valahogyan fel kell állítanunk egy sorrendet, és ezt a hálózati eszközökkel is közölni kell. A DSCP alapján erre egy 0-tól 63-ig terjedő sávunk van, azaz minél magasabb értéket választunk, annál erősebb lesz az adott házirend.

Miután ezzel végeztünk, sorban következik a korlátozandó alkalmazások megnevezése – de akár arra is van lehetőség, hogy az összes alkalmazásra nézve kötelezőnek állítsuk be a házirendet, a forrás és az esetleges cél IP-tartományok (IPv6 is lehet), illetve a protokollok és portok megadásával. Több teendőnk nem is lesz, a házirend a Csoportházirend frissítési ciklusának megfelelően érvényre jut.




*A prioritás és a konkrét sávszélességértékek együtt*

## Terminálszolgáltatások

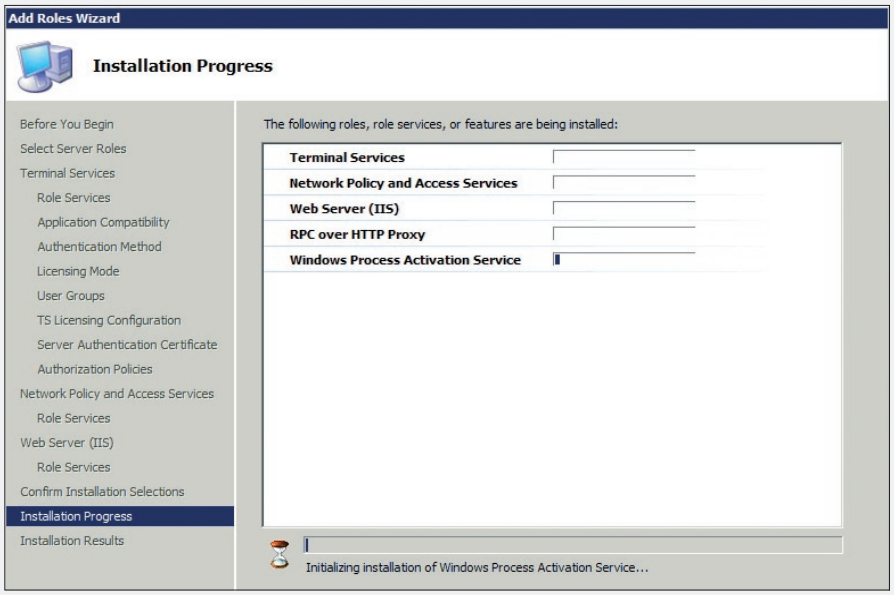
Ha számba vesszük, hogy ezen a területen mennyi változás és újdonság jelent meg már a WS08 bétaverzióiban is, olyan érzésünk lehet, hogy a TS-t fejlesztő csapat valamilyen speciális ajzószert használ saját teljesítményének fokozására. Nézzük csak meg a listát:

- Terminal Services Gateway;
- Terminal Services RemoteApp;
- Terminal Services Web Access;
- Terminal Services Printing.

Sőt, anélkül, hogy itt részletesebben megemlítenénk, a sorba beletartozik a TS Session Broker (ami a helyes újrakapcsolódásért felel, és immár a sessionöket load balancinggal is képes elosztani a TS-farm gépei között!) és az új, a Vistában debütált (XP SP2 és W2K3 SP1 esetén is letölthető) 6.0-s kliens is.

## Terminal Services Gateway

A TS Gateway az SSTP-hez hasonlóan szintén tűzfalbarát megoldás. Lehetővé teszi a távoli felhasználóknak a belső hálózati TS-szerverhez való kapcsolódást HTTPS-en keresztül



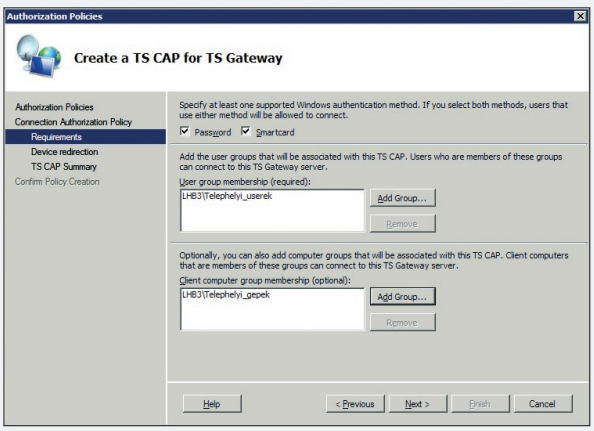

*A TS Gateway szerep telepítéséhez szükséges lépések és kötelező komponensek (a Server Manager szerencsére síkit, ha kimarad valamelyik komponens)*

(RDP over HTTPS). Persze mindezt önmagában képes megvalósítani, azaz például külön VPN-kapcsolat nélkül. Ez azt jelenti, hogy a távoli felhasználónak csak az új RDP-kliensre lesz szüksége a biztonságos kapcsolódáshoz.

Szerveroldalon viszont szükséges ehhez egy Windows Server 2008-kiszolgáló, ami akár a tűzfal is lehet (az ISA-val remekül képes együttműködni), de lehet a DMZ-ben vagy a privát hálózatban is. Így vagy úgy, a lényeg, hogy a távoli kliensnek ehhez a szerverhez kell kapcsolódnia SSL-lel, hogy aztán ez a szerver végezze a konverziót a hagyományos, belső, csak RDP-vel operáló TS-szerver felé.

De nem csak ennyit tesz hozzá a TS Gateway a lehetőségekhez: hozzáférés-szabályzást, illetve erőforrás-elérést is képes ellátni. Nézzük tehát e kétfelé oszló szabályzás lehetőségeit kicsit részletesebben:

- Engedélyező házirendeket (TS CAP = Connection Authorization Policies) gyárthatunk a TS Gateway Manager MMC-ben – vagy akár már a komponensek telepítése közben –, amelyekkel felhasználóknak, csoportoknak adhatunk kapcsolódási lehetőséget a TS Gateway-hez. Mindezt a helyi felhasználói adatbázisból, az AD-ből, illetve ezek hiányában akár teljesen saját készítésű fiókok hatókörében is képesek vagyunk megoldani.
- Olyan házirendet is készíthetünk, amelyben azokat a kliensgépeket jelöljük meg, amelyeknek adunk hozzáférést, de a különböző eszközök (meghajtók, vágólap, nyomtatók stb.) átirányítását is elvégezhetjük közvetlenül a CAP-házirendekből.




*A kapcsolódásnál a felhasználókat és a gépeket tekintve is lehetnek megkötéseink*

- Készíthetünk erőforrás-elérési házirendeket is (TS RAP = Resource Authorization Policy), amelyekkel szabályozható, hogy a hálózaton belül mely gépeket (számítógép-csoportokat) érhetik el távoli felhasználók (akiket szintén szelektálhatunk itt is).

Szintén lehetséges a meglévő helyi vagy AD-csoportok, illetve a helyben, a TS Gateway Managerrel létrehozott csoportok használata, vagy akár úgy is beállíthatjuk, hogy ne legyen semmilyen korlátozás.

- Jó, ha tudjuk: amíg nem hozunk létre mindkét típusból legalább egyetlen házirendet, addig nincs semmilyen hozzáférésre mód a TS Gatewayen keresztül!

A TS Gateway még sok más érdekes és fontos lehetőséget is nyújt, mi most csak az alapokat mutattuk be. Lépjünk tovább a listában, és nézzünk meg két rokonszolgáltatást.

## TS RemoteApp és TS Web Access

A TS RemoteApp komponenssel könnyedén és látványosan megoldhatjuk a terminálkliens-alkalmazás „ellátását" és használatát. A klasszikus felállás szerint a vékonykliensről – szemmel is látható módon – egy RDP-kapcsolatot kell kiépítenünk, majd az adott kapcsolaton (ablakon) belül futtatják a felhasználók a számukra engedélyezett alkalmazásokat.

Az új módszer szerint az üzemeltető (több publikálási módszer közül választva) parancsikonokat helyez el a felhasználó gépén, amelyekre kattintva az alkalmazás egy TS-kapcsolatot kezdeményez a háttérben, ennek hatására elindul a kiszolgálón az adott program, amit a felhasználó úgy vesz észre, hogy a kezelőfelülete rögvest megjelenik a gépén, tökéletes helyi alkalmazásnak „álcázva" magát. Ha a gép vagy felhasználó számára több TS-alkalmazást publikálunk ezzel a módszerrel, akkor a már élő RDP-kapcsolaton testvériesen megosztoznak a háttérben.

Nézzük meg most dióhéjban az üzemeltető oldaláról szükséges teendőket. Rögtön az elején le kell szögeznünk, hogy ez a szolgáltatás csak az RDP 6.0-klienssekkel működik együtt.

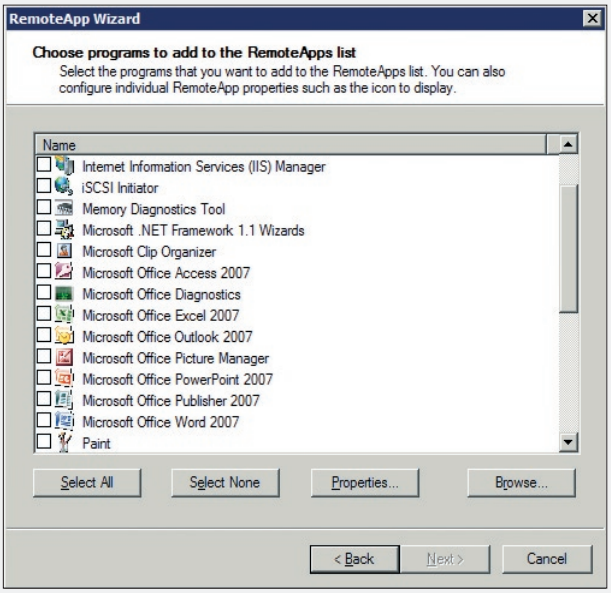
1. Telepítenünk kell a publikálandó programokat, majd ellenőrizni a TS-jogosultságokat és az RDP 6.0 beállításait – mindezt központilag a TS-szerveren.

2. TS Remote App-teendők

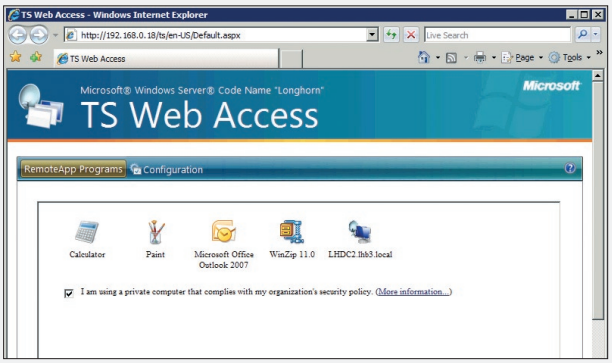
- Az adott program terminálprogrammá „avatása" egy varázslóval (előredefiniált listát kapunk, amelyet persze tetszés szerint bővíthetünk).
- Az adott program terminálprogramként való működésének engedélyezése.
- Egy .rdp vagy .msi csomag elkészítése a kijelölt alkalmazás(ok)ból szintén varázslóval, pár egyszerű lépésben.
- A csomagok publikálása a felhasználók számára (parancsikonok kiszórása, megosztott mappa, Csoportházirend, stb.).

3. A TS Web Access konfigurálása (opcionális)

- szükség esetén a TS Web Accesst is telepíthetjük, amely az IIS7 alatt működve a böngészőkből teszi lehetővé ennek az új módszernek az alkalmazását – kicsit másképp, de hasonlóan hatékonyan és legalább ilyen látványosan.




*Az ezen a gépen publikálható alkalmazások listája*




*Az alkalmazások elérhetők akár a böngészőből is!*

## Terminal Services Printing

Ellentétben az előző TS-alkotóelemekkel, amelyek már a Beta 1 idején is rendelkezésre álltak (persze változtak azóta), ezeket a speciális, nyomtatással kapcsolatos lehetőségeket a Beta 3-ban használhatjuk először.

Az újdonságok két fő részből állnak, az úgynevezett Terminal Services Easy Print meghajtóprogramból és a hozzá tartozó Csoportházirend opciókból.

A speciális meghajtóprogram egy bármilyen RDP-kapcsolat esetén (RemoteApp, Web Access is) lehetővé teszi, hogy a kliens a saját – bármilyen típusú – nyomtatóját minden további beállítás és egyéni meghajtóprogram telepítése nélkül használhassa.

A Csoportházirend-opciók pedig a finomhangolást szolgálják, ezek segítségével tudjuk először is engedélyezni ezt az opciót elsődleges módszerként, illetve óvatosságból azt is megszabhatjuk, hogy csak a kliensen lévő alapértelmezett printert használhassuk automatikusan.

A rossz hír az, hogy ez jelenleg csak akkor működik, ha a kliens egy Windows Vista SP1 (ebben a jelenlegi ismereteink szerint egy még újabb RDP-kliens lesz), és szükség van a .NET Framework 3.0 SP1-re is (a végleges Vista SP1-ben ez már előretelepítve benne lesz). Szerencsére az XP SP2 és a W2K3 SP1 is használhatja majd ezt a szolgáltatást, de csak a Windows Server 2008 RTM kiadásakor lesznek elérhetők az ehhez szükséges komponensek.

A végére érve a legfontosabb megállapításnak az látszik, hogy a Windows Server 2008 hálózattal kapcsolatos szolgáltatásainak, komponenseinek listája terjedelmes méretű, hiszen a cikkben (a néha igencsak szűkszavúan) említett témák mellett az IPv6-ról, a teljesen újraírt TCP/IP-ről (Next-Generation TCP/IP stack, NDIS 6.0), a Scalable Networking Packról vagy akár a Network Policy and Access Serverről sem esett szó.

Gál Tamás
(v-tagal@microsoft.com) Microsoft Magyarország

# Egy fürt Windows

Egy új termékről szólva illendő ötvözni az újdonságok felsorolását azok technikai részleteinek áttekintésével. Mindezt lehetőleg úgy, hogy az olvasó – aki minden témában újdonságokat kap nagy mennyiségben – ne érezze azt nyomasztónak.

A Windows szerverek életében régóta (WolfPack) jelenlévő hibatűrő fürtözéshez minden verziónál kaptunk kiegészítéseket, módosításokat, sőt előfordult, hogy műsoron kívül is jelentek meg új funkciók. A Windows Server 2008 valóban jelentős és már régen várt változásokat hoz az érem mindkét oldalán, legyen az az adminkonzol vagy akár a hálózatkezelés. Az emberiség számítógépekkel foglalkozó tagjai számára régóta fontos kérdés, hogy miképpen lehetne egy szervezet működéséhez fontos alkalmazások, adattartalmak elérhetőségét folyamatosan biztosítani. Amíg valamely érintett komponensből egy van, addig annak meghibásodása kritikus az egész rendszer szempontjából (Single Point of Failure, SPoF), úgyhogy ezeket az elemeket többszörözni kell.

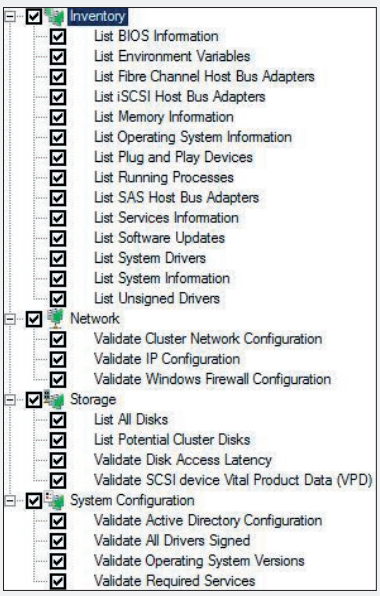
Erre szolgálnak például a különböző RAID-szabványok, amelyek segítségével a tárolás szintjén lehet redundáns és hibatűrő konfigurációkat kialakítani vagy akár a memóriaelemek, tápegységek, hálózati interfészek számát növelni. A failover clustering (átkapcsolásos, esetleg feladatátvételi fürtözés) esetén egyrészt megkíséreljük az összes kritikus elemet többszörözni, másrészt ezeket megfelelő módon vezérelni a magas rendelkezésre állás érdekében.

Egy virtuális pont mögé eldugva több valóságos gép azonos cél érdekében dolgozik együtt, hogy a rajtuk futó alkalmazások és az azok mögött lévő adatbázisok lehetőleg minél tovább hozzáférhetőek legyenek. Az éppen aktív tag meghibásodása esetén a kiszolgálás automatikusan átmozog egy másikra a fürtön belül (amíg még van rá lehetőség), így a külső szemlélő számára folyamatosnak látszik a működés. Ehhez persze arra is szükség van, hogy az adatok bármely tag számára hozzáférhetőek legyenek az adott pillanatban, aminek biztosítására több megoldást is felkínálnak a fürtszolgáltatások.

Fontos a fürt megbízható elérhetőségét is biztosítani, részben a már említett több hálókártyás (team-elt) módszerrel és a hálózati elemek redundanciájával, illetve többrétegű modellekben kombinálható egy másik fürtözési technológia (hálózati terhelésmegosztás, NLB) nyújtotta előnyökkel.

## Telepítés

Egy failover clusternek nagy valószínűséggel fontos szerepet szánunk – magas rendelkezésre állásról és hibatűrésről van szó –, mindenképpen ajánlott tehát az új Validate Tool tesztjeit lefuttatni a telepítés előtt. Ezek tartalmazzák a tagok, a hálózat(i konfig)- és a tárolók tesztjeit, mint azt a mellékelt *1. ábra* is mutatja, tiszteletet parancsoló mennyiségben.

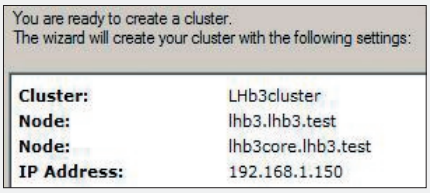



*1. ábra. A telepítés előtti tesztek teljes listája*

Amennyiben megfelelő a tesztek eredménye, kezdődhet a fürt telepítése. Minden platform (x86, x64, IA64) és mindkét szerverváltozat támogatja a fürtözést (a failover és az NLB is), tesztként egy GUI-s és egy Core verziót illesztettünk össze. A grafikus felülettel bíró Windows Server 2008 tényleg nagyon barátságosan és gyorsan konfigurálható, a Core már egy kicsit keményebb dió volt, mert valóban csak parancssora van. Némi kapirgálás után lett fix IP-cím (netsh int ipv4 set address), kicsit később be lehetett rá RDP-zni (cscript c:\windows\system32\scregedit.wsf /ar 0), és máris ott volt ugyanaz a command prompt, mint a konzolon...

Bár mostantól a clusternek nem kell saját szervizfiók, mert LocalSystem alatt fut, továbbra is tartományba integrálva működik, tehát a GUI-s WS08 szerverünkből tartományvezérlő lett, és a Core-t be kellett léptetni a tartományba (netdom join). Ezek után jöhetett a cluster-szerviz installja (start /w ocsetup FailoverCluster-Core), ennek eredményeképpen lett egy, a *2. ábrán* látható konfigunk, majd következett néhány kör a diszkek csatolásával Virtual Serverben a leendő fürt alá. Ez sem volt haszontalan, mert bizonyítást nyert a WS08 cluster szervizének egy fontos változása a Parallel SCSI-csatolókkal kapcsolatban (mivel nem támogatott). Viszont az új szerverek tartalmaznak iSCSI initiatort, és a dolog virtualizációban is átjárható – csak egy jó targetet kell találni – , úgyhogy ez lett a megoldás. A grafikus felüle-

ten viszonylag gyorsan ment az iSCSI-összekapcsolódás, a Core-on el kellett indítani az initiatort (sc \\<server_name> config msiscsi start= auto), majd megint egy újabb parancs (iscsicli) várt felfedezésre.




*2. ábra. Máris készíthetjük a clustert*

Fontos cél volt a fejlesztés során egy jól használható kezelőfelület kialakítása, ahol minden, a clusterhez kapcsolódó műveletet el lehet érni. Innen indultunk a legelején a validációs tesztekkel, és a hardver előkészítése után végül a Create Cluster Wizardban egyszerre kijelölve a két tagot, meglepő gyorsasággal elkészült a fürt, amelyről igen részletes összefoglalás készül a folyamat végén.

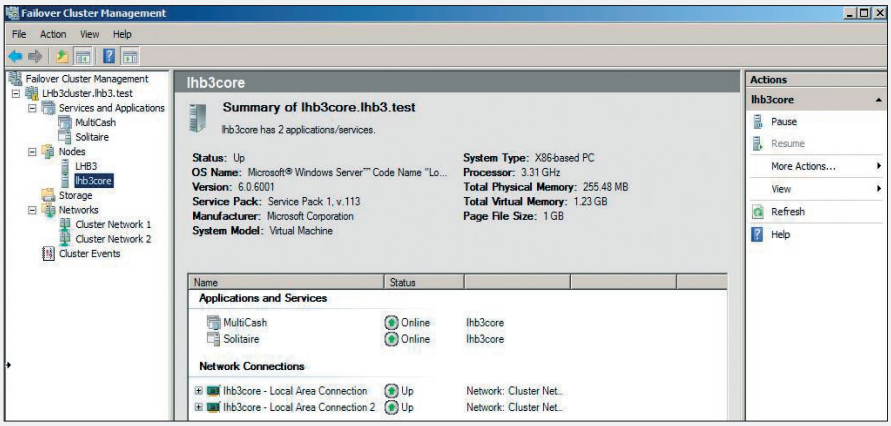
## Adminisztráció

Az új adminkonzol a következő ábrán látható, szintén sok újdonságot mutat, intuitív formában közli az információkat, inkább az alkalmazásokra fókuszálva. A felület három részre oszlik, bal oldalon a szokásos fastruktúra, középen egy infopanel az éppen kijelölt komponens adataival, aktív beavatkozási lehetőségekkel. Jobb oldalon pedig a taszkpanel mutatkozik, szintén dinamikusan alkalmazkodva az objektumokhoz.

```
Microsoft-Windows-ServerCore-Package
Not Installed:BitLocker
Not Installed:BitLocker-RemoteAdminTool
Not Installed:ClientForNFS-Base
Not Installed:DFSN-Server
Not Installed:DFSR-Infrastructure-ServerEdition
Not Installed:DHCPServerCore
Not Installed:DirectoryServices-ADAM-ServerCore
Not Installed:DirectoryServices-DomainController-ServerFoundation
Not Installed:DNS-Server-Core-Role
    Installed:FailoverCluster-Core
Not Installed:FRS-Infrastructure
Not Installed:Microsoft-Windows-RemovableStorageManagementCore
Not Installed:MultipathIo
Not Installed:NetworkLoadBalancingHeadlessServer
Not Installed:Printing-ServerCore-Role
    ¦
    ¦--- Not Installed:Printing-LPDPrintService
    ¦
Not Installed:QWAVE
Not Installed:ServerForNFS-Base
Not Installed:SNMP-SC
Not Installed:SUACore
Not Installed:TelnetClient
Not Installed:WindowsServerBackup
Not Installed:WINS-SC
Not Installed:NetworkLoadBalancingHeadlessServer
Not Installed:Printing-ServerCore-Role
    ¦
    ¦--- Not Installed:Printing-LPDPrintService
    ¦
Not Installed:QWAVE
Not Installed:ServerForNFS-Base
Not Installed:SNMP-SC
Not Installed:SUACore
Not Installed:TelnetClient
Not Installed:WindowsServerBackup
Not Installed:WINS-SC
```

*3. ábra. Mindjárt kész?*

A naplózáshoz kapcsolódóan érdemes megemlíteni az Event Tracing for Windows komponenst az események közötti összefüggések vizsgálatára. Bár sokan nem kedvelik a wizardokat, a konzolból indítható varázslók is van emelve (az összes tagról összeszedegetve és fontosság szerint rendezve).




*4. ábra. A cluster konzolja is beköltözött a 3-as MMC-be*

Érdekesség, hogy a Windows Server 2008-ban naplófronton a hagyományos System–Application–Security triumvirátuson túl is van élet, például a FailoverClusteringnek is saját logja van az Event Viewerben, sőt, ha jobban megnézzük a konzolfa alját, oda is ki láthatóan fejlődtek, jól használhatóak. Ettől függetlenül hozzáférhetők a hagyományos lehetőségek is a fürt kezelésére, lekérdezésére, például a cluster parancs vagy a bővített paraméterkészletű WMI és a Cluster MOM Management Pack. Itt érdemes megemlíteni a

### Az iSCSI

Az iSCSI – szabványos hálózati protokoll az SCSI-forgalom átvitelére TCP/IP-hálózatokon – jelentősége a gigabit ethernet elterjedésével nőtt meg: olcsóbb alternatívája lehet a Fiber Channel-megoldásnak, mivel nincs feltétlenül szüksége saját speciális interfészre, és széles körű kompatibilitást biztosít. Ha ragaszkodunk a megszokott kliens–szerver terminológiához, akkor az initiator az iSCSI-kliens és a target az iSCSI-szerver. Az initiator TCP/IP-hálózaton csatlakozik a targethez, amelyen található erőforrások (lemez, magnó, optikai meghajtó stb.) a kliensen helyi SCSI eszközként jelennek meg, így egyszerűsítheti például fürtök létrehozását is. A gyártói támogatás széleskörű és minden bizonnyal tovább fog bővülni, a Microsoft 2003 júniusa óta rendelkezik initiator modullal, ennek jelenleg 2.03-as változata szabadon letölthető, illetve a WS08 tartalmazza azt. Kis kitartással Windowson futó free targetet is találni – például ilyen a Starwind vagy a MySAN –, komolyabb kihívások esetén a Windows Storage Server R2 szerezhető be.

mentési lehetőségek bővülését: továbblépett a Volume Shadow Copy Service, ezentúl kezeli a fürt erőforrásait is (Cluster VSS writer), és hardveres snapshotok is visszaállíthatók a cluster diszkjeire (Maintenance mode).

## Infrastruktúra és rendelkezésre állás, tárolókezelés

A fürtkonfigurációs lehetőségek terén megmaradtak a már megszokott formációk, a közös quorum és a majority node (MNS) modellek, és lehetséges ezek hibrid alkalmazása páros számú tag esetén is, illetve speciális topológiák támogatására (Node and Disk Majority/Node and File Share Majority), hogy valóban ne legyen a fürtben SPoF. Lehetőség nyílik aktív alkalmazások melletti diszkbővítésre és a függőségek menet közben történő módosítására.

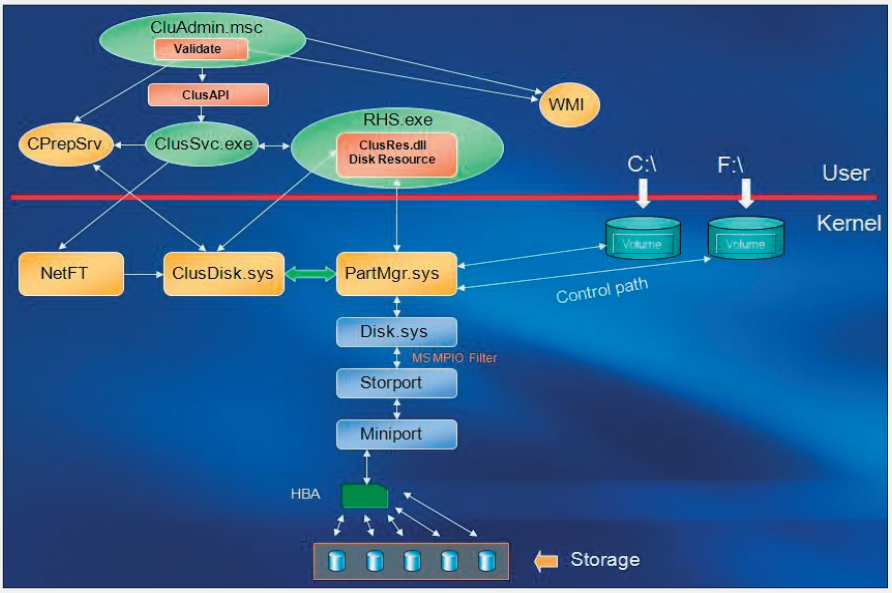
A közös tárolón nem támogatott a parallel SCSI, helyette a SAS, az iSCSI és a Fiber Channel kerül előtérbe, és fejlesztések történtek a SAN-kommunikáció kezelésére, ilyen például a SCSI bus reset elhagyása.

### Simplicity, security, stability

A fenti szavakkal ajánlja a Microsoft új clustering-megoldását. Cluster-telepítés egy menetben, pilótavizsga nélküli fürtmenedzsment – legyen egyszerű(bb) létrehozni és fenntartani bárkinek magas rendelkezésre állású, hibatűrő szolgáltásokat a hálózatán.

Hát tessék, akkor így múlik el a világ dicsősége, mostantól varázslóval bárki telepíthet magának clustert?

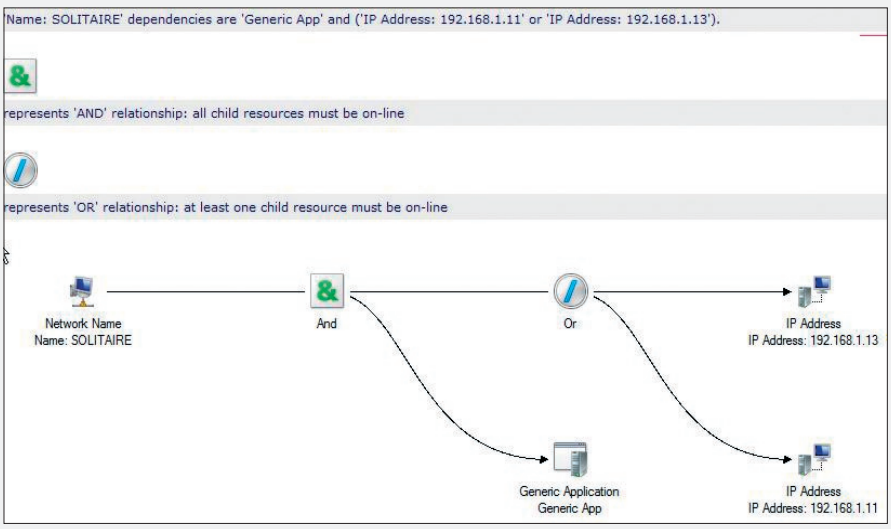
Természetesen nem, mivel a szoftveren kívül a fürt három H-s ügy: kell hozzá jó **Hardver**, stabil **Hálózat** és inkább több, mint kevesebb **Hozzáértés**, ilyenkor pedig egy wizard – legalábbis ebben a verzióban – legfeljebb asszisztálni tud.




*6. ábra. A Windows Server 2008-clusterek architektúrája*

## Hálózatkezelés

Jelentős előrelépés a fürt kiterjeszthetőségében, hogy az új cluster-szerviz megengedi a különálló alhálózatokon lévő tagok összekapcsolását extra VLAN konfiguráció nélkül (a heartbeat TTL-je 1-re volt állítva, és ennek köszönhetően nem tudott eljutni egy másik subnetbe). Szükség esetén átállítható a heartbeat timeoutja is, így összeköthetők a földrajzilag távoli szerverek is, vagy a csökkentéssel lerövidíthető a hibák észleléséig eltelt idő. Megszűnik a NetBIOS függőség a névfeloldásban (plusz az ilyen irányú broadcastforgalom is) és DNS alapon megy tovább. A kommunikáció a Windows Server 2008 IP-stackjére épül, természetesen teljes körű IPv6-kompatibilitással a fürtön belül a tagok között – illetve a kliensek és a fürt kommunikációjában. Tisztán Kerberos-autentikáció és semmi NTLM, UDP helyett TCP. Nem hanyagolható el a függőségek kezelésében történt változás sem, egy Network Name erőforrás aktív maradhat, amíg az alatta lévő IP-címek közül legalább egy elérhető.




*5. ábra. Dependency Report*

## Konklúzió

A gyors áttekintés végén hozzá kell tenni, hogy a lista még messze nem teljes. Az tisztán látható, hogy komoly fejlesztésekkel és fontos változásokkal találkozunk a szolgáltatásban a gyártó által célként megjelölt négy területen (security, networking, eventing, storage), és mivel a beta3-as már feature complete verzió, ezekkel a leírt formában biztosan számolhatunk. A téma kiegészítéseként érdemes elmélyedni a következő linken elérhető további részletekben is:

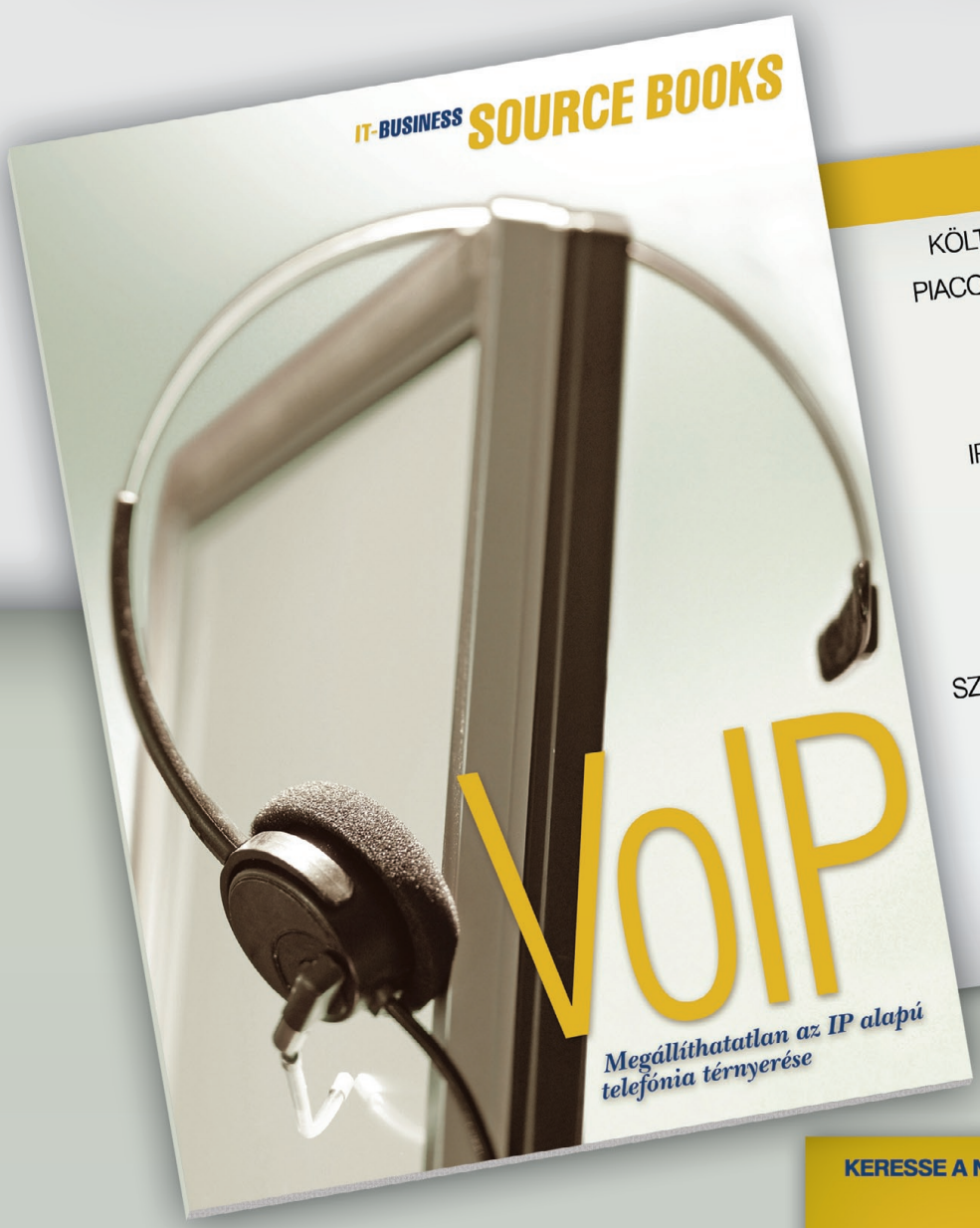
http://www.microsoft.com/windowsserver/longhorn/failover-clusters.mspx.

*Ország Tamás*
MCSE+S+M, MCT
(tamas@edupro.hu) Számalk

IT-BUSINESS SOURCE BOOKS

# VoIP

## ESZKÖZÖK, MEGOLDÁSOK, BIZTONSÁG




- KÖLTSÉGEK ÉS FUNKCIÓK
- PIACOK, TRENDEK, SZÁMOK
- VOIP ÉS SKYPE
- KOCKÁZATOK ÉS MEGOLDÁSOK
- IP-TELEFONRENDSZEREK
- HAGYOMÁNYOS TELEFONOKBÓL VOIP-KÉSZÜLÉK
- PROBLÉMÁK ÉS KÖVETELMÉNYEK
- SZABVÁNYOK ÉS BIZTONSÁG
- … ÉS MÉG SZÁMOS ÉRDEKES TÉMA

**MAGÁNSZEMÉLYEKNEK, VÁLLALATOKNAK ÉS KORMÁNYZATI SZERVEKNEK EGYARÁNT!**

ÁRA: 4000 FT

**KERESSE A NAGYOBB KÖNYVESBOLTOKBAN, VAGY RENDELJE MEG ONLINE**

**MEGRENDELÉSI LEHETŐSÉGEK:**

**TEL.: 888-3421, 888-3422**
**FAX: 888-3499**

**ONLINE: WWW.IT-BUSINESS.HU**
**E-MAIL: TERJESZTES@VOGELBURDA.HU**

# Erős bástya

## A Windows Vista biztonsági újdonságai mind beépültek a Serverbe is. Felmerülhet a kérdés: vannak-e további újdonságok?

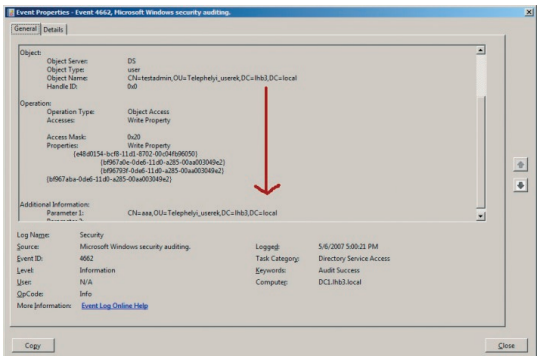
Gyakorlatilag költői a kérdés, azonnal el is oszlatjuk az esetleges kételyeket: sok-sok újdonsággal állunk szemben – ahogyan az várható is volt. Főképpen azért, mert a Vistával közös megoldások a rendszerek alapszintű működését érintik (UAC, házirendek, BitLocker, az IE védett módja stb.), a Windows Server 2008 viszont egy hálózati, kiszolgáló-operációsrendszer, olyan nagy és jelentős megoldásokkal és komponensekkel, mint például a címtár, a tanúsítványkiszolgáló infrastruktúra vagy éppen – teljes újdonságként – a hálózati hozzáférést fizikai szinten ellenőrző eszköz.

### Biztonsági változások a címtárral kapcsolatban

Több idetartozó témát már tárgyaltunk a TechNet Magazin Active Directoryról szóló cikkében (a talán legnagyobbat, azaz a RODC-t is), de azért maradt még jó pár téma, amelyek közül két nagyobb lélegzetűről be is tudunk számolni. Az első az auditálásban bekövetkezett érdekes és fontos változás, amely lehetővé teszi, hogy jobban nyomon kövessük az AD-objektumok változásait. Bekapcsolhatunk például olyan új audit-házirendet, amellyel lehetőségünk nyílik a címtárszolgáltatások naplózásánál az adott objektum régi és új értékét egyaránt megfigyelni.

A Windows 2000/2003-kiszolgálókban egyetlen audit-házirend van, amellyel engedélyezhetjük vagy tilthatjuk a címtárral kapcsolatos hozzáférések naplózását (az 566-os számú eseményekben látszik majd ennek az eredménye) – de ez a módszer egyrészt nem túlságosan bőbeszédű, másrészt nem túlságosan testre szabható. A Windows Server 2008-ban ez a házirend négy részre lett szétszedve:

- Directory Service Access,
- Directory Service Changes,
- Directory Service Replication,
- Detailed Directory Service Replication.




*Az objektum névváltozásának követése*

Az alkategóriák bekapcsolásához az auditpol parancssori eszközt kell használni (a GUI-n ebből nem sok látszik – legalábbis egyelőre). Ha viszont egyszer bekapcsoltuk, akkor lehetőség lesz az adott objektum (mondjuk egy OU), biztonsági jellemzőin keresztül az Audit szakasz alá felvinni egy-egy bejegyzést, amelynek következtében aztán a Security-naplóban láthatjuk majd a kapcsolódó események pontos leírását, illetve a változások előtti és utáni állapotot (a 4662 ID-ra kell szűrnünk). Lássunk néhány egyszerű példát a jellemzők részletességének naplózására:

- az objektum bármely jellemzőjének megváltoztatásakor a régi érték megjelenítése (usernév, telefonszám stb.);
- egy objektum tartományon belüli mozgatásakor a régi, illetve az új lelőhely megnevezése;
- egy objektum törlésből történő visszaállításakor a visszaállítás helyének megnevezése.

Jó tudni azt is, hogy ez az új lehetőség rendelkezésre áll majd az AD LDS-nél is (Lightweight Directory Services), azaz az ADAM utódjánál.

Egy másik, alaposan megújult szolgáltatás a tanúsítványszolgáltatások területe. A címtárral történő együttműködés már az elnevezés változásából is kiderül, innentől ugyanis összefoglaló néven Active Directory Certificate Servicesnek hívjuk az idetartozó komponenseket, amelyek a következők:

- ADCS: Web Enrollment,
- ADCS: Policy Settings,
- ADCS: Network Device Enrollment Service,
- ADCS: Enterprise PKI (PKIView),
- ADCS: Online Certificate Status Protocol Support.

A Web Enrollment komponens változott talán a legtöbbet, ami például azt is jelenti, hogy az ActiveX alapú megoldást a Vistában és a Windows Server 2008-ban lecserélték egy új és egyszerűnek tűnő COM alapú eszközre. De nem kell megrémülnünk az új eszköz (CertEnroll.dll) mellett a régi szerverekkel (illetve egy-két megkötéssel a WS08-kiszolgálókkal is) működhet a régi webes tanúsítványbeszerző és -megújító eszköz (Xenroll.dll), ráadásul az áttéréshez a WS08-ban több lehetőség is rendelkezésre áll.

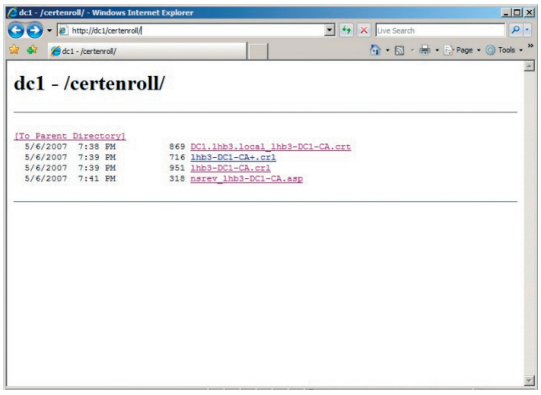
Az idevágó rengeteg új házirend-opcióval (ADCS: Policy Settings) is érdemes lenne foglalkozni, de talán még ennél is érdekesebb a Network Device Enrollment szolgáltatás, amely a Microsoft implementációja a Simple Certificate Enrollment Protocol (SCEP) szabványprotokollra. Az NDES lehetővé teszi, hogy a különböző hálózati eszközök (szoftverei) hitelesítési célokra x509-es tanúsítványokat igényelhessenek, például a saját CA-szerverünktől, így többek között IPSec-kapcsolatokat könnyebben tudunk majd létesíteni ezekkel az eszközökkel. Az NDES ISAPI filterként működik a CA szerver IIS-én, így az IIS képes lesz:

- generálni és kiosztani az egyszeri igényléshez szükséges admin-jelszavakat;
- elkapni és feldolgozni a routerek, switchek által „benyújtott” SCEP-igényléseket;
- a CA felől pedig visszairányítani a függő kéréseket (Pending requests).

A (korábbi) PKIView-komponens sokak számára ismerős lehet már – eddig a Windows Server 2003 Resource Kit részeként volt használható –, mostanra viszont beépült a Windows Server 2008-ba, egy MMC bővítmény formájában, és új nevet is kapott (Enterprise PKI). Egy ráncfelvarrással is átesett, és így a komplett, vállalati rendszerünk PKI-infrastruktúráját, illetve a CA-nk felügyeletét is képes kezelni, mert:

- egyetlen hierarchikus nézetben képes megjeleníteni a PKI-infrastruktúrát és a címtárhoz való viszonyát;
- érti és használja a Unicode-karaktereket;
- szülő/gyerek nézetben képes bemutatni a CA-k hierarchikus fastruktúráját;
- az MMC-ből minden csomópont közvetlenül felügyelhető;
- a különböző állapotokra használt színkódok segítségével gyorsan megítélhető a tanúsítványok, a CA-k, és a PKI-rendszer egészének állapota.

A következő komponens viszont teljes mértékben újdonság. A hagyományos tanúsítvány-visszavonási listák (Certification Revocation List, CRL) használata mellett a WS08-kiszolgálókban lehetőségünk van az Online Responder MMC modul használatára, azaz az Online Certificate Status Protocol (OCSP) előnyeinek kiaknázására. Ezek az előnyök akkor jelentkeznek, ha például egy távoli bejelentkezéskor a kliens nem képes letölteni egy nagyobb méretű CRL-t, vagy ha időnként komoly méretű terhelés alatt áll a hálózatunk a CRL-ellenőrzés miatt (sok egyidejű hálózati belépés, e-mailek vizsgálata stb.), vagy például akkor, ha egy nem Microsoft CA-tól kapott visszavonási listát kell „szétszórni” a tartományban.




*http://szerver_neve/CertEnroll – így is lehet*

## Network Access Protection (NAP)

A NAP valószínűleg a Windows Server 2008 legnagyobb – biztonsággal kapcsolatos – „dobása” lesz. A legtöbb szervezet esetén ugyanis jelentős igény mutatkozik egy olyan megoldásra, amely már a fizikai hálózat szintjén elválasztja az alkalmi csatlakozású vagy kevésbé megbízható, illetve kevésbé felügyelhető számítógépeket a belső hálózatba tartozó, értékes kliensektől és szerverektől. Tartományi környezetben van néhány eszközünk a biztonsági határok felállítására, a központi felügyeletre és a renitens gépek „móresre tanítására”, de sok esetben még ez is kevés. Viszont a fizikai hozzáférés szintjén egyáltalán nem rendelkezünk ezekkel az eszközökkel. Ugyanakkor nagyon sok esetben nem tagadhatjuk meg teljesen a hozzáférést a fizikai hálózatra szükségszerűen jogosan kapcsolódó (nem tartományi) gépektől sem.

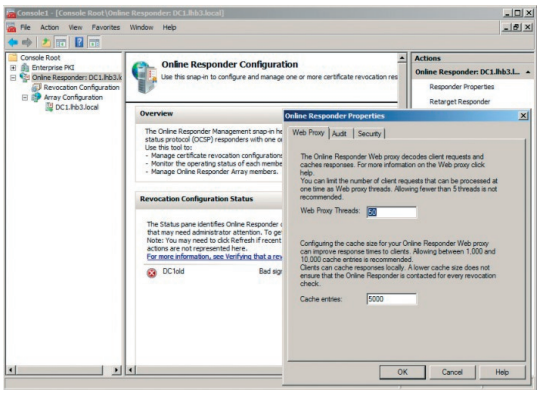
Erre a láthatóan nehezen megoldható helyzetre nyújthat gyógyírt a NAP, azaz egy olyan szerver–kliens megoldás, amely a védett hálózatunkban alapértelmezés szerint még az IP-kapcsolatot sem engedi meg, és amely csak egy alapos, az üzemeltetők által részletesen hangolható „vizsga” sikeres teljesítése esetén adja meg a hozzáférést a belső hálózathoz kapcsolódni szándékozó gépeknek.

A NAP célja tehát az, hogy a routerek, switchek, a vezetéknélküli hozzáférési pontok, a szoftveres és az appliance (hardverbe épített célszoftver, például egy ISA Server 2006) rendszerek segítségével érvényesítse a végpontbiztonságot. Mindezt úgy éri el, hogy lekérdezi a hálózatra csatlakozó eszközök biztonsági állapotát (bekapcsolt tűzfal, AU-kliens, vírus/spyware-irtó állapota stb.), majd ezt összehasonlítja az előre definiált biztonsági szabályzattal, és az eredmény alapján dönt a hálózati hozzáférés engedélyezéséről. Ha a folyamat negatív eredménnyel zárul, a kapcsolódni szándékozó kliens nem jut be a védett hálózatba, hanem lehetőséget kap biztonsági állapotának „szintre hozására”, azaz csatlakozhat a publikus szegmensen működő WSUS/SMS-szerverhez, a vírusirtó-szignatúrákat tároló szerverhez, majd a szükséges korrekció után végrehajthat egy újabb kapcsolódási kísérletet.

A tisztánlátás kedvéért említsük meg a NAP egyetlen előzményének tekinthető – már a Windows Server 2003-ban és az ISA 2004/2006 kiszolgálókban is jelenlévő – VPN-karanténmegoldást, de csak azért, hogy kiderüljön: mélységében és használhatóságában egyaránt nagyon különbözik a NAP-tól. A legjobb példa erre a beléptetőközeg, ugyanis a VPN-karantén értelemszerűen csak a VPN-kapcsolatok esetén volt használható, míg a NAP a vezetékes, a vezetéknélküli és a RAS/VPN-kapcsolatok esetén is, illetve a kapcsolódás típusa alapján a DHCP-, IPSec- vagy éppen az RDP-kliensekre is alkalmazható.

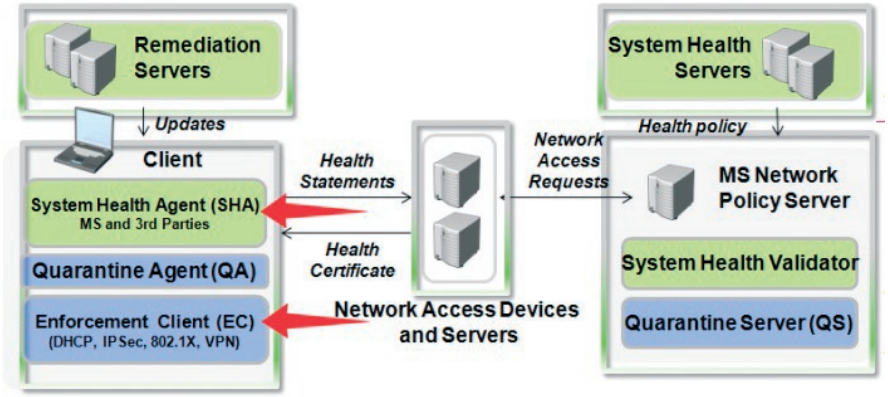
Fontos azt is tudni, hogy az első NAP-kiszolgáló a Windows Server 2008 lesz, kliensoldalon viszont a Vista már tartalmazza a megfelelő összetevőket. Egy jó ideje elérhető – de egyelőre csak a tesztelők számára – az XP SP2-re telepíthető NAP-kliens is.

Ha szándékunkban áll a NAP bevezetése, akkor számoljunk azzal is, hogy több kiszolgáló-szerepkör is felkerül majd a kiszolgálónkra. A függőségek miatt nem kell aggódnunk, mert a Server Manager korrekt módon megoldja az összes szükséges elem feltelepítését.




*Az Online Responder web proxy cache konfigurálása*

**Network Policy Server (NPS).** A Microsoft legújabb RADIUS-szerverimplementációja. Az NPS segítségével központilag képesek leszünk felügyelni a vezetéknélküli hozzáférési pontokat, a VPN-szervereket, a dial-up szervereket és például a 802.1x szabvánnyal dolgozó switcheket. Sőt, az NPS használható biztonságos jelszóhitelesítésre vezetéknélküli hálózatok esetén – a PEAP-MS-CHAPv2 protokollal.




*A NAP-hierarchia*

**NAP Policy Server.** Az NPS egyik üzemmódja, amellyel a kliensek „egészségi” állapotát felmérő és az eredményt tároló csomagok (Statements of Health – SoH) kézbesíthetők.

Az NPS-t használhatjuk RADIUS-proxyként is. Ekkor a kapcsolódási kérések továbbküldhetők a megfelelő kiszolgálóra.

**RRAS.** Szintén felkerül a szerverre ez a régi jó ismerős is, amely a sokféle kapcsolódási lehetőségért, illetve hálózati technológiáért felel (VPN, DUP, LAN-to-LAN, LAN-to-WAN, NAT stb.)

**Health Registration Authority (HRA).** Egy olyan NAP-komponensről van szó, amely csak az IPSec típusú kapcsolódáskor szükséges. A kliens (ami ebben az esetben tipikusan valamilyen hardvereszköz lesz) egy – a megfelelt állapotát bizonyító – tanúsítványt kaphat ettől a kiszolgálótól, az SoH-kérés kiküldése után.

**IIS7 + PKI-infrastruktúra.** Szintén kötelező elemek, már a telepítővarázslóban is konfigurálhatunk egy saját CA-t (persze ha már létezik, akkor használni is tudja a megfelelő szervertanúsítványt).

**Host Credential Authorization Protocol.** A Cisco hasonló (NAC, Network Admission Control) rendszere felé kapcsolatot teremtő komponens.

Ezek után tisztázzunk egy-két ismeretlen fogalmat, illetve további elemet, amelyek a NAP-hierarchia részei (az első három fogalom a kliensoldalra vonatkozik):

**System Health Agent (SHA).** Feladata a Vista/XP rendszer alkalmasságának ellenőrzése, majd ennek az információnak a jelentése.

**Enforcement Client.** A kliens kapcsolódási metódusát jelöli. Minden hálózati hozzáférési típushoz egy-egy külön NAP EC áll a rendelkezésre. Ha például a klienseink a DHCP-vel kapcsolódnának a NAP-kiszolgálóhoz, akkor ezt kell engedélyeznünk.

**NAP-ügynök.** Az EC-k és az SHA közötti információcsere bonyolítója.

**System Health Validator (SHV).** Az SHA-któl érkező jelentéseket (SoH) ellenőrzi, és egy válasszal (Statement of Health Response) jelzi, hogy a kliens megfelel-e vagy sem.

**Health Policy.** A korlátlan és a korlátozott hozzáférések kritériumait írja le. A fenti Enforcement Clienteknek megfelelően több – IPSec-, DHCP- stb. – szabályzat is létezhet.

**Accounts Database.** Az adott hálózat felhasználóinak és számítógépeinek tárolója, ami a legtöbb esetben a címtár lesz.

**Remediaton Servers.** A „gyógyító” szerverek, azaz amelyekhez a kliensek fordulhatnak a biztonsági állapotuk megfelelő szintre emelésének céljából.

Végül nézzük meg, hogy mi történik egy működő NAP rendszerben, amikor a kliens DHCP-vel szeretne kapcsolódni.

1. Vistánk a szokásos módon IP-beállításokat kér a DHCP-kiszolgálótól.

1.1. Ha a kliens rendelkezik az SoH-val, akkor ezt a DHCP-kérelem már tartalmazza. A DHCP-szerver az SoH-t továbbítja az NPS-kiszolgálónak, az pedig a Policy Servertől megtudakolja, hogy érvényes-e.

„A” eset: ha igen, a DHCP-kiszolgáló ellátja a klienst a komplett IP-konfigurációval, a kliens pedig korlátlan hozzáférést kap a hálózathoz, és vége a folyamatnak.

„B eset”: ha nem, akkor a DHCP olyan IP-paramétereket ad át a kliensnek, amelyek csak a korlátozott hálózathoz biztosítanak hozzáférést.

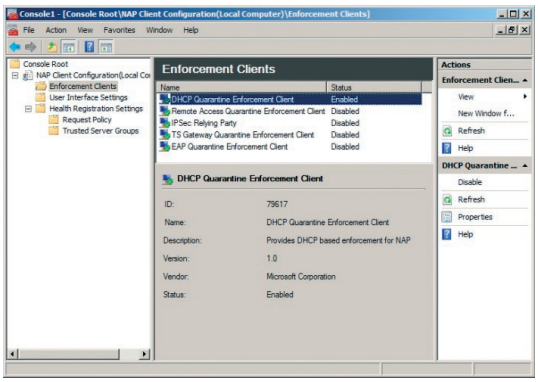
1.2. Ha a kliensnek nincs SoH-ja, akkor nem megfelelő, és ugyanaz történik, mint az előző pont „B” esetében, és jön a 2. pont.

2. A kliens NAP-ügynöke frissítési kérelmet küld a nyitott hálózaton lévő javítószervernek.

3. A javítószerver ellátja a klienst a megfelelő javításokkal, és a kliens SoH-ja is aktualizálódik.

4. Ismételt kérelem indul a DHCP-szerver felé, amely viszont már tartalmazza az új SoH-t – azaz kezdődhet elölről a vizsgálat.

A Windows Server 2008 biztonsági megoldásai a fenti témakörökkel természetesen nem merülnek ki, hiszen – többek között – a BitLocker, az IPSec, a Windows-tűzfal, az RMS-kiszolgáló, az EFS újdonságairól szó sem volt, de természetesen a különböző TechNet-eseményeken ezek ismertetésére is sor kerül majd.




*A NAP-kliens beállításai – történetesen egy WS08-on*

*Gál Tamás*
(v-tagal@microsoft.com) Microsoft Magyarország
*Kelemen László*
(kelemen@hungary.com)

# IT-SECURITY TODAY

INFORMATIKAI BIZTONSÁGI HAVILAP NAPI ONLINE TÁJÉKOZTATÓJA

- informatikai döntéshozóknak, technológiai szakembereknek
- az elmúlt 24 óra legfontosabb hazai és külföldi informatikai biztonság és információbiztonság hírei
- ingyenes napi online hírlevél


## Regisztráljon!

www.it-business.hu/hirlevel

# Internet Information Services 7

Az Internet Information Services még az NT4-ben kezdte hosszú pályafutását, 1996-ban. Most már a hetes változatnál tartunk, ami a Windows Vistában debütált, és benne lesz a Windows Server 2008-ban is – ezt a változatot tekintjük át cikkünkben.

Már az NT4 alatt újabb és újabb verziókkal találkozhattunk szervizcsomagok formájában. A 3-as IIS-változatban jelent meg az Active Server Pages- (ASP-) támogatás, ami nagyon sikeres lett, hisz a platformon ekkor lehetett először egyszerűen dinamikus tartalmat szolgáltató webalkalmazásokat fejleszteni. Utólag, fejlesztőként nézve, már van mit kritizálni rajta, de akkor ez tényleg nagy szám volt. Az Option Packnek nevezett csomagban már egy egészen használható webszerver volt, az IIS4. El lehetett szigetelni a webalkalmazásban futó kódokat az IIS-től, ami megbízhatósági szempontból alapvetően fontos volt (úgyis bugos kódot írunk, készül rá a webszerver). A Windows 2000-rel jött az IIS5, ettől kezdve az IIS összeforrt egy-egy operációsrendszer-verzióval. Az IIS5 finomítása volt az előző verziónak, nem volt benne észveszejtő újítás, inkább teljesítmény- és megbízhatósági hangolást végeztek rajta.

Ekkor jött a drámai bukás. A biztonsági rések miatt rendszeresen hekkelték az IIS4/5-öket, ami jelentős presztízsveszteséget okozott a Microsoftnak. A cég azonban kitartóan alkalmazza a nietzschei elveket: „Ami nem öl meg, erősebbé tesz”.

A Windows 2003-hoz kapcsolt IIS6 fejlesztése során brutális biztonsági kódátnézéseket vezettek be *Michael Howard* vezényletével. Ekkor indult a Microsoftnál „Trustworthy Computing” belső és külső kampány, ami egyes cégek sokszor nagyképű és üres kampányával szemben valóban ért is valamit. Az IIS6-ot kutyaerősre edzették, az eddig eltelt 4 év alatt csupán 3 hibát találtak benne, és ezeket sem lehetett távolról kihasználni. Hihetetlen, hogy nagyon erős elszántsággal az egyik biztonsági szempontból legrosszabb visszhangot kiváltó terméket ennyire jóra átírtak. Az IIS6 már sokkal gyorsabban tudta kiszolgálni az ASP.NET-alkalmazásokat is, mivel architekturálisan felkészítették rá, szemben az IIS5-tel, ami még az ASP.NET 1.0 előtt jött ki.

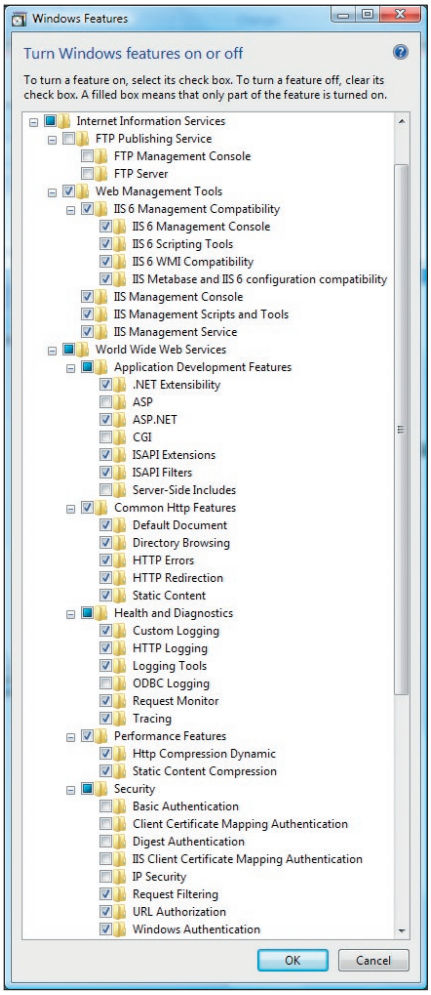
Egyszóval az IIS6 nagyon jól sikerült webszerver lett. De akkor mi marad az IIS7-re? Lássuk!

## Modularizált felépítés

Az IIS7 a Vistában debütált, és ez lesz a Windows Server 2008-ban is, csak addig még tovább analizálják és javítják a biztonság és a teljesítmény szempontjából. A legfontosabb újítás architekturális jellegű: darabokra szedték szét a szervert, modularizálták. Miért fontos ez? Az IIS6 többek között azért volt sokkal biztonságosabb elődeinél, mert alapban csak nagyon kevés szolgáltatás volt engedélyezve rajta. Például tegyük fel, hogy ASP.NET-lapok kiszolgálására használtuk a szervert. Miért kellene ekkor bekapcsolva lennie a klasszikus ASP-futtatónak? Mi van, ha pont a nem használt részben van egy biztonsági rés, és azon keresztül törik fel a kiszolgálót?

Az IIS7-ben nem kikapcsolva vannak a nem használt szolgáltatások, hanem fel sincsenek telepítve a gépre! Azaz nem is lappanganak a gépen DLL-szinten, várva arra, hogy a hacker engedélyezze őket, majd visszaéljen vele.

Azt mondhatnánk erre, hogy ez nem nagy szám, át kellett írniuk a telepítőt, hogy több mindent lehessen fel-le rakni, és kész. A helyzet azért nem ilyen egyszerű. Aki írt már nagyobb lélegzetvételű programot, az tudja, hogy a szoftver komponensei között függőségek vannak. Ha A nincs fenn, B se fog menni. Nagyon tudatos tervezéssel és nagy fejlesztési fegyelemmel lehet csak azt elérni, hogy egy szoftvert szét lehessen szedni na-




*1. ábra. A moduláris felépítésű IIS7*

gyon sok pici darabra. Nos, az IIS7 közel 40 darabra esett szét, ezeket egyesével lehet ki-be kapcsolni a telepítés során *(1. ábra)*.

Nem kell Basic-autentikáció? Akkor ne is legyen a gépen, így biztos nem kapcsolják be véletlenül se rendszeradminisztráció során vagy rosszindulatúan távolról. Soha, senki sem használta a Digest hitelesítést? Akkor miért eszi a memóriát a kódja? Kiderül, hogy bug van a CGI-futtatóban? Na és, fel sincs rakva. Nem kell frissíteni a fel sem telepített komponenst, így a szerver nem áll le még pár perce se, nem kell újraindítani a hotfix miatt, és még a családunk is látni fog este 8-kor.

## Konfigurációkezelés

A korábbi IIS-ek egyik bosszantó fogása volt, hogy a konfigurációs adatokat ügyesen eldugta az orrunk elől a MetaBase-nek nevezett adatbázisába. Ebben az volt a „jó”, hogy bináris formátumú volt, így ha behülyült (jaj, de értett hozzá még az IIS4), és nem volt mentésünk, akkor lehetett újraépíteni a website-okat, virtuális könyvtárakat, egyebeket. Hamar megtanult mindenki MetaBase-t menteni.

Az IIS6-ban már XML formátumú volt a MetaBase, volt nagy öröm. Azonban az egyik gond az volt vele, hogy a rajta leggyakrabban futtatott ASP.NET-nek megvolt a saját konfigurációja machine.config vagy web.config fájlokban, míg a webszerver konfigurációja teljesen máshol, a MetaBase.xml-ben volt. Lehetett keresgélni, mi hol van.

Ha valaki nem is ASP.NET-futtatásra használta az IIS6-ot, akkor is gonddal járt a konfigurációs feladat szétosztása, mivel a MetaBase egy nagy, centralizált XML állomány volt, így nem lehetett könnyen delegálni például egy adott website konfigurációját bizonyos embereknek.

Pedig az ASP.NET-ben már ki volt dolgozva egy hierarchikus, elosztott, web.config alapú konfigurációs rendszer, csak maga a webszerver-konfiguráció maradt monolitikus.

Nos, az IIS7-ben nemcsak a futtatható komponenseket modularizálták, hanem a konfigurációkezelést is. Egy minimális központi konfig persze maradt, 32 bites gépeken a %windir%\System32\InetSrv\config\ApplicationHost.config fájl.

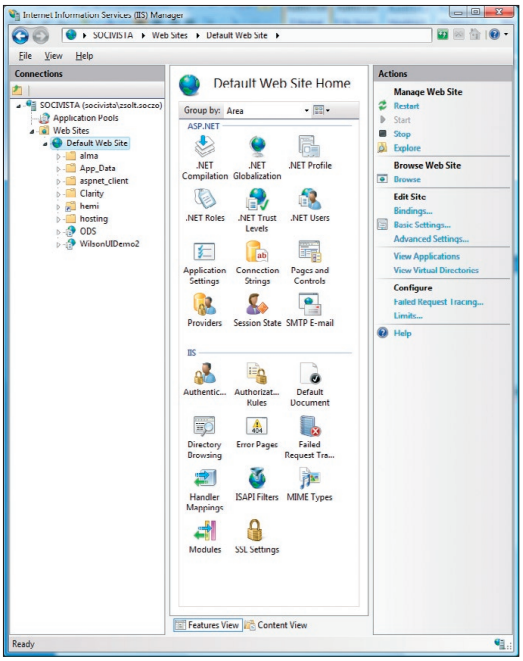
Erre szükség van, mert valahol csak le kell írni, milyen website-ok és milyen application poolok (processzek, amelyek betöltik és futtatják a webalkalmazások kódját) vannak a szerveren, illetve vannak egyesével nem állítható jellemzők, mint a naplózás vagy a HTTP-tömörítés, ezeket továbbra is csak egy helyen lehet konfigurálni. Ezek az ApplicationHost.config system.applicationHost szekciójában laknak.

Másrészt itt vannak az alapbeállítások, amelyeket az új website-ok örökölnek, de szabályozhatóan átállíthatnak – a system.webServer szekció. Így már külön állományokban lehet konfigurálni az egyes website-ok jellemzőit, elosztottan (egyesével szabályozható, mit lehet felülírni alacsonyabb szinten), ráadásul nem kell adminnak lenni a gépen, csak, hogy legyen joga valakinek a saját kis site-jának vagy könyvtárának web.config-ját szerkeszteni. Ez fontos dolog, nem kell mindenért rendszergazda után kiáltani.

## Adminisztráció

Az elosztott konfigurációs XML állományokat nem kell feltétlenül kézzel szerkeszteni, van hozzá szép GUI is *(2. ábra)*. Mint látható, az egymás hegyén-hátán lakó tulajdonságlapok már nem divatosak, helyette klasszikus Control Panel-szerű retróstílus mellett döntöttek az IIS admin tervezői (szólni kellene az Active Directory Users and Computers szerzőinek, sűrűbben kellene találkozniuk az IIS grafikusával).

Ami nagyon jó, hogy szemben a korábbi IIS Managerrel, nem RPC-n keresztül megy be a webszerverre, ami miatt tésztaszűrőt kellene csinálni a tűzfalból, hanem HTTPS-en keresztül, amihez egyetlen apró lyuk is elég a tűzfalon, és „mellesleg” még biztonságos is.




*2. ábra. A konfigurációs felület*

Sőt, a hitelesítéshez felhasználható az ASP.NET Membership felhasználói adatbázisa is, így az adminisztrációnak nem kell Windows-felhasználóhoz kötődnie.

Aki programozottan szerette volna az IIS-t adminisztrálni, az általában kis scriptkódokat írt, és ez persze sok rendszergazdának nagyon életidegen tevékenység volt. IIS7-ben kapunk egy appcmd.exe-t, ami meglepően sokat tud, így jól használható batch-fájlokban. A parancsok nagyon jól paraméterezhe-

tők, a célobjektumok keresésekkel jól megragadhatók.

Például rögtön egy egzotikummal kezdve az éppen futó kérések listáját láthatjuk (a'la SQL Server Profiler):

```
C:\Windows\System32\inetsrv>APPCMD list requests
REQUEST „fc0000008000001c" (url:GET /
WilsonUIdemo2/, time:3266 msec, client:localhost)
REQUEST „fa00000080000004" (url:GET /
WilsonUIdemo2/, time:3204 msec, client:localhost)
```

Vigyázzunk, hogy valóban rendszergazdaként futtassuk a parancsot, különben furcsa hibát kapunk, aminek semmi köze a jogosultság hiányához.

Aki tényleg programozottan szeretné konfigurálni az IIS7-et, az használhatja a felügyelt kódú adminisztrációs objektumokat a Microsoft.Web.Administration névtérből vagy a WMI-interfészt.

Zseniális, hogy hibák esetén nemcsak egy hibakódot lök az ember arcába a szerver, hanem javaslatot is ad a megoldásra. Például, megszokásból felépítettük a web.configot egy modul betöltésére:

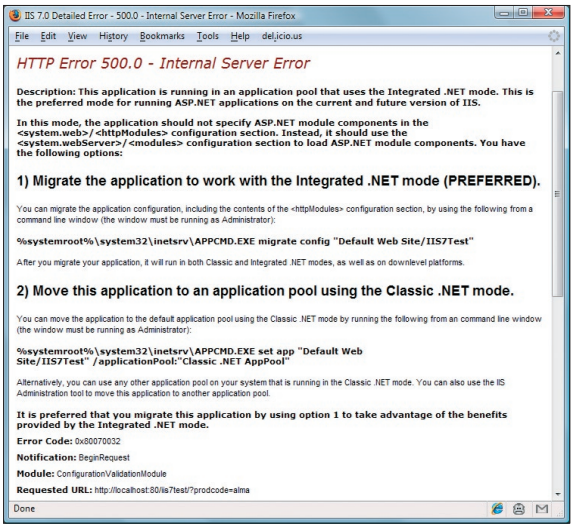
```
<system.web>
 <httpModules>
  <add name="Rewiter"
    type="IIS7Test.UrlRedirectModule"/>
 </httpModules>
</system.web>
```

Erre jött egy igen részletes hibaüzenet *(3. ábra)*, amelyben leírják mi a gond, és milyen alternatív lehetőségeink vannak a hiba feloldására. Ott a kész parancs a probléma feloldására, már csak le kell futtatni! Ez tetszik.

## Programozási felület

Az IIS tulajdonképpen egy nagyteljesítményű, többszálú HTTP-protokollkiszolgáló. Az összes plusz tudása – mint az ASP.NET-lapok futtatása – külső modulokban van megvalósítva. IIS6-ban a statikus fájlkiszolgálás még a szerver része volt, az IIS7-ben még ezt is külső modul valósítja meg. Ezeket a modulokat tradicionálisan az ISAPI nevű interfész felhasználásával implementálták, C++-ban. Ilyen volt az ASP-futtató is, de a .NET integráció is egy ISAPI DLL-ben volt implementálva. Ez utóbbiból jól látszik, hogy az ISAPI interfész a legközvetlenebb csatlakozási pont a szerverhez, míg például az ASP.NET is csak alkalmazása ennek. Ez amellett, hogy frusztráló lehet a .NET-programozók számára, egyes esetekben valódi gondokat okozhat.

Az egyik gyakori feladat például statikus tartalom, html, pdf, képek stb. védelme az




*3. ábra. Beszédes hibaüzenetek és megoldási javaslat*

ASP.NET Forms hitelesítésével. Sajnos ez nem olyan egyszerű IIS7 előtt, mivel csak egyes kiterjesztéseket – mint aspx, asmx stb. – dolgozott fel az ASP.NET-futtató, a többit az IIS direktben szolgálta ki. A nem ASP.NET-fájlokat is lehet védeni, de ahhoz át kellett konfigurálni az IIS-t, hogy az egyéb fájlkiterjesztések is az ASP.NET-motorba fussanak be. E kérések kezelése nem minden esetben egyszerű vagy kivitelezhető.

Az IIS7-ben létezik egy új feldolgozási üzemmód, amelyet Integrated Mode-nak hívnak. Ebben már az ASP.NET ugyanolyan képességekkel rendelkezik, mint az ISAPI-alkalmazások, így például egy ASP.NET modul minden további nélkül szabályozhatja a hozzáférést bármely nem ASP.NET-fájlhoz. Megszűnt az IIS–ASP.NET kettősség, a kettő valóban teljes mértékben egybeolvadt. Integrált módban egymás után lehet kötni például egy natív C++-modult, valamint egy ASP.NET-modult, és mindkettő dolgozhat az összes fájltípuson, nemcsak a maroknyi regisztrálton.

Kompatibilitási okokból a régivel azonos körülmények között működtethetők a régebbi alkalmazások a Classic üzemmód alatt.

A teljes ASP.NET-integráltság ellenére az ISAPI interfész továbbra is rendelkezésre áll, ráadásul jóval egyszerűbb lett, és sokkal biztonságosabb a használata. Miért volt eddig veszélyes? Mert natív kódról beszélünk, kézi memória- és erőforrás-menedzsmenttel, amelyben elég könnyű elfeledkezni egy-egy lefoglalt memóriablokk felszabadításáról, vagy handle lezárásáról. Közönséges alkalmazásoknál ebből nincs mindig dráma, de szerverek esetén, ahol naponta sok ezer vagy millió kérést szolgálnak ki, a hibák hatásai összeadódnak, és a szerver a programozó karmájának áldozata lesz.

Az új ISAPI interfészben az IIS allokálja és szabadítja fel a memória zömét, így kisebb felelősség hárul a fejlesztőkre, kisebb a valószínűsége a memóriát szivárogtató programoknak. Emellett már nem sík függvényekkel, C alapon kell programozni, hanem objektumorientáltan, C++-osztályokkal, így a saját adataikat is egyszerűbb kezelni az egyik legfontosabb erőforrás-szervezési C++-alapelv, a Resource Acquisition Is Initialization (RAII) felhasználásával.

## Biztonság

Mint már szó volt róla, az ASP.NET-modulok működnek nem ASP.NET-hez rendelt tartalomra is, így az ASP.NET Forms hitelesítése bármilyen tartalomra használható, anélkül, hogy minden kérést az ASP.NET-motorra kellene terelni. Így nem kell kézzel kezelni a statikus fájlok és a dinamikus tartalmak kiszolgálását, mégis védhetők az erőforrások a Forms hitelesítéssel. Ez igen hasznos, mert így a védett statikus fájlokat jóval gyorsabban lehet kiszolgálni, mint ha mindent az ASP.NET-motor kezelne.

Az IIS7 kapott két szerviz fiókot az anonymous kérések megszemélyesítésére: egy IIS_IUSR felhasználót és egy IIS_IUSRS csoportot. Mindkettő rögzített SID-del rendelkezik, azaz minden gépen ugyanazzal a SID-del szerepelnek. Ez azért jó, mert így a fájlokra adott jogosultságok átvihetők másik gépre, nem kell az ACL-eket migrálni. Tesztgépről végleges szerverre költözéskor,

illetve webfarm új gépének beállításakor nagyon hasznos, hogy most már van az IIS-nek beépített felhasználója.

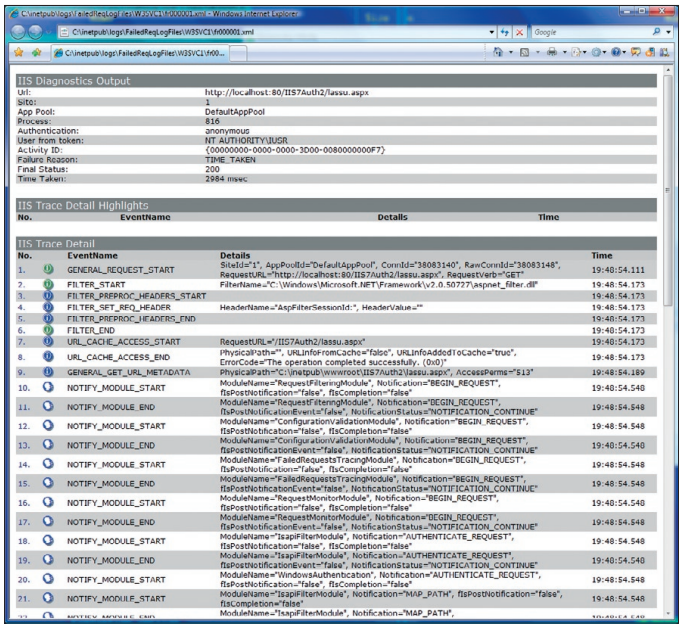
További hasznos jellemzője a beépített felhasználónak, hogy nincs jelszava, így az nem jár le, illetve nem esik ki a szinkronból a valódi jelszó és az IIS által ismert.

Az eddigi URLScanhez hasonló beépített szolgáltatás a Request Filtering, amellyel korlátozható a kérések, illetve az URL-ek hossza stb. Egy újabb védelmi szint a hackerek ellen. Eléggé aluldokumentált, de az applicationHost.config requestFiltering ágának vizsgálata, illetve a %windir%System32\inetsrv\config\schema\IIS_schema.xml segíthet felfedezni.

## Diagnosztika

Mivel a webkiszolgálók a háttérben futnak, és többszálú környezetben hajtják végre a kéréseket, néha igen nehéz kitalálni, mitől áll le a kiszolgálás. Nem triviális hibára kell gondolni, amikor kivétellel elszáll egy lap, azt amúgy is központilag szokás kezelni és a megfelelő értesítéseket elküldeni. Sokkal aljasabb hibák keletkeznek, amikor valamiért a worker threadek összeakadnak vagy leállnak. Bár sok szál fut egyszerre, ha egy bugos kód megállít egyet-kettőt deadlock vagy valamilyen hosszú idejű folyamat miatt, és ugyanez a kérés ismétlődő módon befut a szerverhez (a legtöbb kérés nyilván ilyen), akkor pár perc alatt leállíthatjuk az összes szálat.

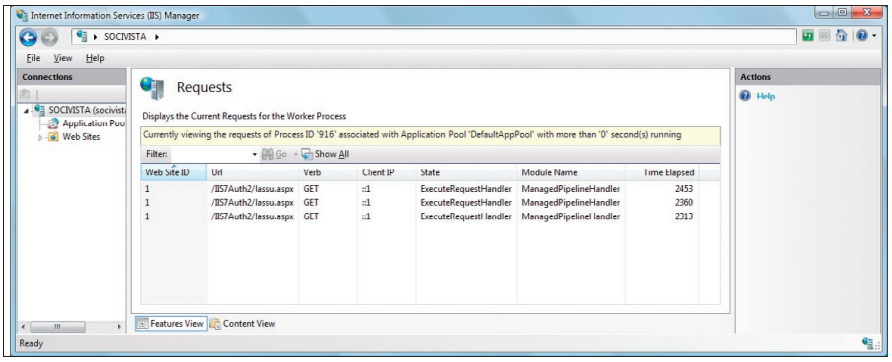
Egyszer egy rosszul megfogalmazott reguláris kifejezés miatt állt le egy lap, de csak bizonyos QueryString paraméterek mellett. ASP-lapként implementálva csak az ADPlus segítségével sikerült kinyomozni, mely lapok és mely paraméterek mellett volt beteg, és egyáltalán, melyik lap okozott gondot.




*5. ábra. Egy lassú kérés nyomai*

ASP.NET alapon már lehetett írni olyan modult, amelyik megmutatta, mely lapok végrehajtása tartott tovább egy adott időnél, ASP alatt ezt csak ISAPI filterrel lehetett volna megoldani.

Az IIS7 sokat fejlődött diagnosztika terén. Írtak hozzá egy Runtime State and Control API-t, RSCA-at, amit Riskának ejtenek (tényleg). Ezzel amellett, hogy megnézhető az Application Poolok és egyéb komponensek állapota, az aktuálisan végrehajtott kéréseket is láthatjuk. A korábbi appcmd-s példa is erre épít a háttérben, a grafikus felületet kedvelők a *4. ábrán* láthatnak három lassú kérést.




*4. ábra. A futó lekérdezések állapota*

Érdekes látni az éppen futó kéréseket, de aligha valószínű, hogy valaki főállásban szeretné őket sasolni. Csak naplózással egybeépítve igazán hatásos ez a szolgáltatás. Az IIS6-ban a Windows 2003 SP1 részeként már volt egy Event Tracing szolgáltatás, ezt fejlesztették tovább az IIS7-ben. Erre építették fel a Failed Request Tracing Feature-t, rövidítve a FREB-et. Paraméterezhető, hogy milyen feltételek esetén készítsen logot a kérésről. A logok alapban itt szaporodnak, XML formátumban: C:\inetpub\logs\FailedReqLogFiles\W3SVC1\. Az XML-ekhez írtak egy XSLT-t, így Internet Explorer alatt szépen formázva láthatóak a hiba részletes körülményei.

Megadhatjuk, mely HTTP-válaszkódok esetén szeretnénk lenyomatot, például csak a 404-es hibák érdekelnek, azaz a hiányzó erőforrások miatt fellépő hibák. Másfelől azt is meg lehet adni, hogy nem érdekes a státuszkód, de ha a kérés több mint X másodpercig fut, akkor legyen róla információnk. Ezzel lehet kiszűrni a lassan futó lapokat, amelyek blokkolást okozhatnak a szerveren.

Egy lassú kérés nyomát láthatjuk az *5. ábrán*.

A logokat időnként elemezve még a nagyobb baj bekövetkezte előtt ellenlépéseket lehet tenni, például optimalizálni az SQL-lekérdezéseket, mielőtt a látogatók jelzik, hogy használhatatlanul lassú a website-unk.

## Zárszó

Az IIS7 a Vista egyik legszebben felépített komponense. Mivel a Vista asztali operációs rendszer, egyelőre még csak tesztelni, tanulni lehet, majd a Windows Server 2008 lesz az, amiben már élesben is be lehet vetni. Addig is érdemes szemezgetni vele, és a jövőre kibocsátandó alkalmazásoknál már meggondolni az új szolgáltatások kiaknázását.

*Soczó Zsolt*
MCSD, MCDBA, MCT, ASP.NET MVP
(zsolt.soczo@gmail.com)

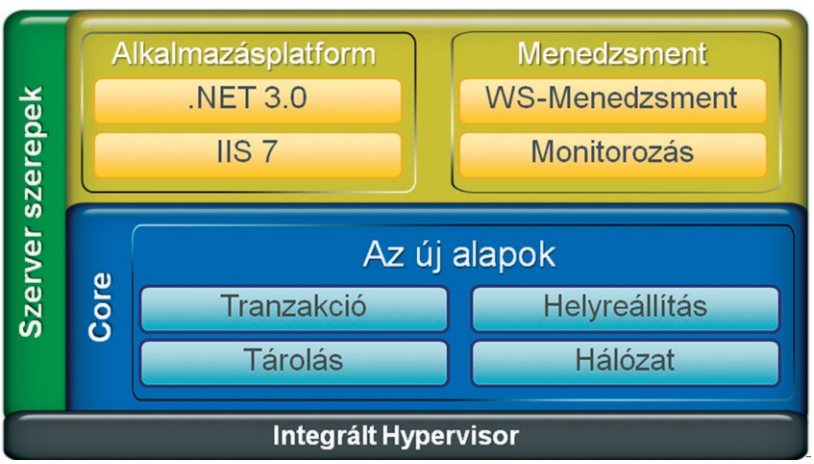
# Windows Server 2008 – egy fejlesztő szemével

Szinte lehetetlen feladat a sok-sok év munkájaként megszülető új szerver fejlesztői újdonságainak rövid bemutatása – most mégis megkíséreljük ezt a legérdekesebb funkciók kiemelésével.

Ismerkedésünket az új rendszerrel érdemes az architektúra blokkvázlatának tanulmányozásával kezdeni. Rögtön látható, hogy a Windows Server 2008 integrált része a .NET keretrendszer hármas változata (és alapból fel is van telepítve) – így bátran hozzáláthatunk a Windows Presentation, Communication és Workflow Foundation vagy a Windows Cardspace alapú fejlesztésekhez. Ha alkalmazás- és webszerverre is szükségünk van, rendelkezésünkre áll az Internet Information Services 7.0, mint a Windows Server 2008 egy opcionálisan telepíthető szerepköre. Szintén az alkalmazásplatformhoz kapcsolódik az MSMQ 4.0-s változata.

## Üzemeltetésre készített szoftver

Menedzsmenttechnológiák terén is jó néhány új képességgel találkozhatunk. Jól látható, hogy a Dinamikus Rendszerek Kezdeményezés a Windows Server 2008-at is elérte, itt különös tekintettel az üzemeltetésre készített szoftver koncepciójára gondolunk. Ennek a megközelítésnek az a lényege, hogy a fejlesztők a szoftvereik elkészítésekor – már jó előre gondolkodva – úgy végzik el a tervezés és fejlesztés lépéseit, hogy a kész rendszer könnyen telepíthető, üzemeltethető, felügyelhető legyen. Ha nem kell saját adminisztrációs felületeket, illetve script-futtató és -értelmező környezeteket építenünk, valamint rendelkezésünkre áll egy jól működő, hibakeresésre fókuszáló platform, akkor máris könnyebb az életünk.




*A Windows Server 2008 fejlesztői újdonságainak blokkvázlata*

Az ehhez szükséges technológiák között előkelő helyet foglal el a Microsoft Management Console 3-as változata, amelyhez mi is készíthetünk saját snap-ineket, ráadásul most már felügyelt, .NET-es kód írásával is. Az MMC 3.0-s felületek alá pedig saját PowerShell-scripteket és providereket is készíthetünk, így kiszolgálhatjuk saját alkalmazásunk és a rendszergazdák igényeit egyaránt. Szintén ide kapcsolódik a továbbfejlesztett eseménynapló (Windows Eventing 6.0) használatának lehetősége, ami már teljesen XML alapú – ez is tökéletesen kezelhető felügyelt kódból, és jól ráépíthetünk automatizáción alapuló, eseményvezérléssel működő megoldásokat az új Task Scheduler API segítségével. Kihasználhatjuk a WS-Management protokollt is, hogy tűzfalbarát módon tudjunk rendszerfelügyelettel kapcsolatos kommunikációt végezni szoftvereink között – erre épül egyébként a WinRM/WinRS páros is, amellyel a távoli parancssoros felügyelet valósítható meg.

Ide kapcsolódik még, hogy kihasználhatjuk a Windows Server 2008 Terminal Services szerepkörének képességeit, és készíthetünk olyan alkalmazásokat is, amelyek felhasználói felülete a kliensen jelenik meg ugyan, mégis minden a háttérben a szerveren zajlik (Remote Programs). Ezzel kapcsolatban számtalan újítás történt, ez leginkább a Terminal Services szerepkörnek és az új Remote Desktopnak köszönhető. Mindezen technológiák rendszergazdák számára érdekes vonatkozásairól már részletesen esett szó a TechNet Magazin korábbi cikkeiben – a legfontosabb változás most az, hogy ezeket a fejlesztők akár saját alkalmazásaik szerves részévé tehetik a jövőben.

## Helyreállítás

A Windows Error Reporting (WER) lehetősége és működése nem újdonság a Windows Server 2008-ban, ellenben az, hogy ezt saját hasznunkra is fordíthatjuk, már igen. A platform részeként elérhető API segítségével saját jelentéseket készíthetünk egyedi hibaesemények bekövetkeztekor (nemcsak akkor, ha leállás következik be), vagy akár egyénire szabhatjuk a hibavisszajelző ablakok felületét, tartalmát is. A hiba feladása után pedig a saját terméktámogató weboldalunkra is irányíthatjuk a felhasználókat, ahol megnézhetik, hogy az előforduló hibáikra született-e már megoldás, illetve, hogy milyen lehetőségeik vannak további segítség igénybe vételére.

A Recovery olyan lehetőség, amellyel az alkalmazás egy kritikus hiba előfordulása esetén még megmentheti a felhasználó vagy a rendszer számára kritikusan fontos információkat. Miután az alkalmazás a megfelelő helyeken beregisztrálta magát, leállás esetén az adott rutinok még az erőforrások (fájlok, memória stb.) eleresztése előtt meghívódnak, és lehetővé teszik az adatok összegyűjtését, elmentését, az erőforrások felszabadítását. Az alkalmazásnak még arra is lehetősége nyílik, hogy a hiba előfordulása után újraindíttassa magát. A Restart Manager segítségével pedig minimalizálhatjuk a telepítések és rendszerváltoztatások következtében szükségessé váló rendszer-újraindítások számát.

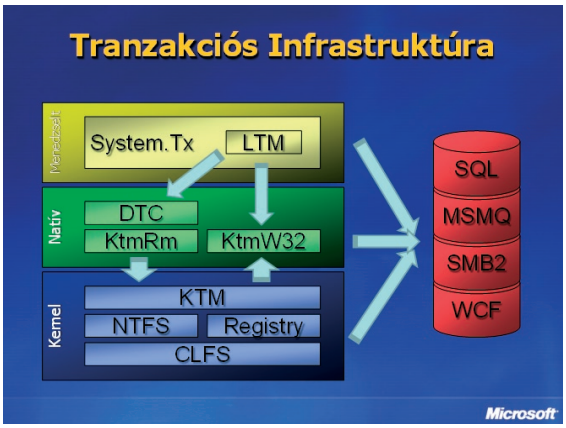
## Konkurenciakezelés

A szálak rendezésével és a szálak csoportjainak kezelésével kapcsolatban is sok újdonságot találhatunk, ezek közül különösképpen azok az érdekesek, amelyek az eseményekre sokat várakozó alkalmazások számára jöttek lére. Az ilyen alkalmazások szálai létrehozásuk után sok időt töltenek várakozó állapotban egy-egy esemény bekövetkeztére. Alkalmazásainkban egyes szálakat időzítetten felébreszthetünk, hogy állapotokat ellenőrizzenek vagy állítsanak be.

A korábbról már jól ismert Thread Pool technológiában megjelenő új API-készlet számos új lehetőséget rejt, mind rugalmasságban, mind a követhetőségben túllépi a régit. Például a hibakeresést nagyban megkönnyíti, hogy a számunkra érdekes szálak akár folyamatosan is nyomon követhetők, vagyis a debuggert utasíthatjuk arra, hogy a szál ne kerüljön várakozó állapotba addig, amíg működését vizsgáljuk. Emellett lényegesen kibővültek az imperszonációs lehetőségeink is.

## Tranzakciós fájlrendszer és registry

Ez egy nagyon érdekes és újszerű téma, érdemes tehát egy kicsit részletesebben is elmerülnünk benne. A tranzakciók kezelését eredetileg az adatbázisokhoz fejlesztették ki, és több, szorosan összefüggő problémára próbált meg választ adni egyszerre. A tranzakciók négy alapvető célkitűzése röviden az ACID-jellemzőkkel írható le. A tranzakciók biztosítják az összetartozó műveletek egyszerre történő érvényesítését (Atomic), a tranzakciók végrehajtása utáni adatkonzisztenciát (Consistent), az egyes folyamatok átmeneti inkonzisztens állapotának más folyamatok előli elzárását (Isolation) és a műveletek eredményének tartósságát (Durability).




*A tranzakciós infrastruktúra a Windows Server 2008-ban*

A tranzakciók használatával lehetőségünk van arra, hogy a rendszerünk teljesítményén, konzisztenciáján vagy konkurenciáján növeljünk – természetesen mindre egyszerre nincs lehetőség, de adatbázis-kezelő rendszereknél mi magunk dönthetjük el, melyik számunkra a fontosabb adott esetben. Ezzel részletesen foglalkozott a TechNet Magazin egy korábbi cikke az SQL Server 2005-re vonatkoztatva.

Felmerülhet a kérdés, hogy miért csak adatbázisok esetén használjuk ezeket a technológiákat? Korábban a Microsoft már lehetőséget adott arra, hogy a Microsoft Distributed Transaction Coordinator segítségével összeköthessük különféle platformok és szoftverek tranzakcióit, azonban a tranzakciók kezelésére továbbra is csak adatbázisszerverek (például SQL Server 2005), illetve rendszerintegrációs megoldások (mint például a BizTalk Server 2006) esetén volt lehetőségünk beépített módon – csak ezek ismerték igazán a tranzakció fogalmát. Természetesen saját fejlesztéseket is kapcsolhattunk a DTC-hez, de ennek korrekt megvalósítása nem minden esetben a legegyszerűbb feladat.

A Windows Server 2008-ban a tranzakciókezelés a Kernel egyik alapképességévé vált (Kernel Transaction Manager – KTM), így bármilyen folyamat vagy változtatás, ami a rendszerben történik, elméletben körbefogható tranzakciókkal. Ott még nem tartunk ugyan, hogy az operációs rendszer valamennyi erőforrása képes legyen ezt kihasználni (például a rendszermemóriát még nem kezelhetjük tranzakcionálisan), de két fontos erőforráscsoport már kihasználja ezt az új lehetőséget. Az egyik a fájlrendszer, a tranzakcionális NTFS (T-NTFS), a másik pedig a tranzakcionális rendszerleíró adatbázis (Transact Registry).

Mindkét megoldás a Common Log File System technológiájára és a KTM-re épít. Maga a KTM jól skálázható a kis alkalmazásoktól a komolyabbakig, biztosítja az alkalmazások, a tranzakciók és a tranzakciótípusok közti izolációt, hogy mindig konzisztens adatokat láthassunk. Az SMB 2.0-s protokoll használatával lehetőségünk nyílik arra is, hogy a fájlműveleteink más gépekre

is átnyúljanak, valamint mindkét új technológia részt tud venni a DTC-s elosztott tranzakciókban.

## A gyakorlatban

Az elméleti síkról váltsunk a gyakorlatra! Konkrétan mi fog másképpen működni ennek következtében a fájlrendszerben és a registryben?

A fájlrendszer és a registry izolációs szintje „Read Committed". Az SQL Server 2005-ös adatbázisok esetében is ez az alapértelmezett izolációs szint, így az azt használóknak ez a működési elv nagyon ismerős lesz.

**Fájl vagy registry-kulcs módosítása.** Gyakori és egyszerű művelet. Persze, ha közben hiba történik az alkalmazásunkban, és emiatt a fájl vagy a registry-kulcs módosítása félbemarad, az eredmény akár katasztrofális is lehet. Az üzletileg kritikus alkalmazások pont ezért végezték eddig ilyen műveleteiket fájlátnevezések és -másolások útján, biztosítva a folyamatos konzisztenciát – de erre most már nincs szükség, hiszen a tranzakcionális fájlrendszer lehetővé teszi az automatikus védelmet. Például ilyen esetekben a változtatás valamennyi lépésének legvégéig az adataink eredeti állapotukban maradnak, és kizárólag akkor fognak megváltozni fájljaink és registry-kulcsaink, ha minden lépés hibátlanul lezajlott.

**Konkurens írás és olvasás.** Az előző pontban leírt művelet kapcsán nemcsak azt nyerjük, hogy nem hagyunk ott magunk után esetlegesen félbehagyott módosításokat, hanem azt is, hogy amíg mi módosítunk az adatokon, addig más alkalmazások továbbra is hozzáférhetnek az eredeti, semmilyen más alkalmazás által végzett tranzakcióban történt módosítással nem rendelkező változathoz. A fájl vagy registry-kulcs írása nem kell, hogy az adott fájl olvasás elől való lezárását eredményezze, az továbbra is olvasható marad a régi tartalommal a tranzakció lezárásáig. A tranzakciós rendszer az erőforrások jellegénél fogva egy tranzakcionált író és akárhány olvasó egyidejű jelenlétét teszi lehetővé egy-egy erőforráson – viszont a tranzakció megnyitása után a nem tranzakcióban lévő írásoknak várniuk kell.

### További információk

Mount Gellért – http://blogs.msdn.com/MountGellert

**Több fájl vagy registry-kulcs együttes módosítása.** Ugyancsak gyakran előforduló művelet, hogy fájlmódosításokat, átnevezéseket együtt kell kezelni, például egy weboldal átnevezését és a többi fájlban az oldalra mutató linkek módosítását. A tranzakcióba történő foglalás lehetővé teszi, hogy hiba vagy közben előforduló más probléma (akár a felhasználó által a művelet megszakítása) esetén gyorsan és egyszerűen konzisztens és helyes állapotba hozhassuk a rendszert.

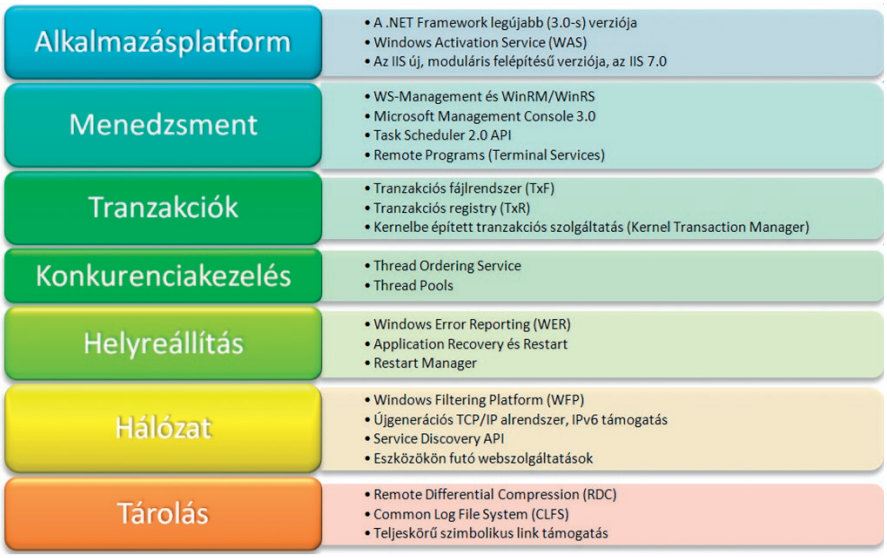
Fontos kiemelni, hogy a tranzakciók használata opcionális, valamint szükség van hozzá alkalmazásaink átírására is. Természetesen használata a tranzakcióban részt vevő fájlok számának és méretének arányában többleterőforrásokat igényelhet, azonban az általa nyert előnyök ezért kárpótolnak minket. A tranzakciók megtervezésekor ugyanúgy, mint az adatbázisok esetén, érdemes a tranzakciós ablakok méretét a lehető legkisebbre venni, hogy minimalizáljuk az erőforrások felhasználását.

Ami még külön érdekes: a használható hardveregységek listája messze túlmutat a szokásos eszközökön, akár arra is lehetőségünk van, hogy NTFS-re formázott USB-meghajtónkat bevonjuk a műveletekbe.

## Hálózat

Továbbra is támogatott lesz a Transport Driver Interface az alacsony szintű kommunikáció programozására, azonban annak leváltására egy lényegesen egyszerűbb és nagyobb tudású Winsock Kernel (WSK) érkezik.

Emellett rendelkezésünkre áll a Windows Filtering Platform (WFP) API is, ami egységes felületet biztosít a hálózati forgalom szűrésére, feldolgozására és transzformálására. Ennek segítségével végre közös platform áll össze, így most már az operációs rendszer is pontosan látja, hogy mely alkalmazás milyen portokon és milyen protokollok segítségével kommunikál. Nem ok nélkül épül erre a technológiára a Windows Vista tűzfala is. Ha WFP-n alapuló alkalmazásokat készítünk, lehetőségünk nyílik arra is, hogy könnyedén hozzunk létre saját tűzfalszabályokat, és kihasználhatjuk az IPSec infrastruktúrát is.




*Rengeteg újdonság érkezik a Windows Server 2008-ban*

Ezen kívül a fejlesztők újratervezték a teljes hálózati stacket is, ez most a Next Generation TCP/IP stack nevet viseli. Legfontosabb előnye, hogy egyaránt és egyformán támogatja minden hálózati szolgáltatás esetében az IPv4 és IPv6 alapú kommunikációt is.

Az újdonságok listáját hosszan folytathatnánk még, hiszen várhatóan vaskos könyvek íródnak majd ezekről, azonban jól látható, hogy számtalan hasznos dolog érkezik a fejlesztők számára a Windows Server 2008 megjelenésével, és ennek csak egy szelete az amúgy önmagában is hatalmas 3-as .NET keretrendszer.

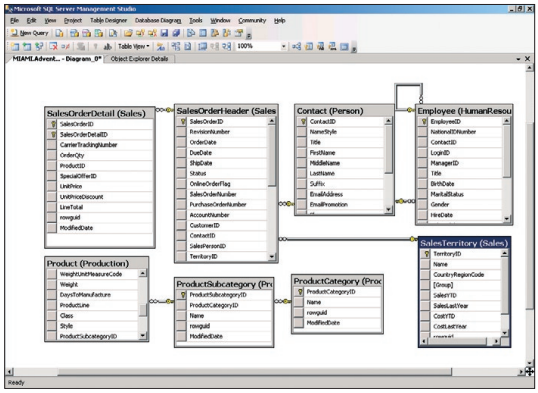
*Smulovics Péter*
(petersm@microsoft.com) Microsoft Magyarország

# OLAP-ALAPOK

## Egy tipikus feladat: összetett kimutatáskészítő rendszert építünk.

Az OLAP (Online Analytical Processing) rendszerek lényege, hogy rajtuk keresztül az adatokat számos elemzési szempont szerint csoportosítva, összesítve mutathatjuk meg a felhasználóknak, akik azokat tovább csoportosíthatják, szűrhetik és különböző eszközökkel online elemezhetik. Az OLAP-modellek szemléletmódja, szerkezete alapvetően eltér az adatok relációs vagy objektum alapú szemléletétől. Cikkünkben egy OLAP rendszer felépítésének lépéseit és a legfontosabb fogalmakat mutatjuk be egy tipikus feladaton keresztül.

Vizsgáljuk meg egy egyszerű példán, hogy miért is érdemes OLAP-modellek építésén gondolkodnunk. Az *1. ábrán* egy megrendeléskezelő rendszer adatmodellje látható.




*1. ábra. Relációs modell*

Tételezzük fel, hogy a cég vezetésétől azt a feladatot kapjuk, hogy készítsünk kimutatásokat termékenként és kereskedőnként az eladásokról, az adatok legyenek lekérdezhetők éves, negyedéves és havi összesítésben, lehessen azokat csoportosítani értékesítési régiónként, és ha lehet, akkor viszonteladónként is. Az igények további elemzése során kiderül, hogy a felhasználók különböző módokon szűrni is akarják az adatokat, például csak a 100 dollárnál olcsóbb termékek eladási adatait akarják látni, vagy egy bizonyos kereskedő teljesítménye érdekli őket. A felhasználók egy része pedig Excelben szeretné tovább elemezni az adatokat, ezért azt szeretnék, hogy a kimutatások közvetlenül az Excelben jelenjenek meg.

```
SELECT
    Sales.SalesTerritory.[Group],
    Person.Contact.FirstName,
    Person.Contact.LastName,
    SUM(Sales.SalesOrderDetail.OrderQty) AS Quantity,
    SUM(Sales.SalesOrderDetail.LineTotal) AS Amount
FROM
    Person.Contact INNER JOIN
        HumanResources.Employee
            ON Person.Contact.ContactID = HumanResources.Employee.EmployeeID INNER JOIN
        Sales.SalesPerson
            ON Sales.SalesPerson.SalesPersonID = HumanResources.Employee.EmployeeID INNER JOIN
        Sales.SalesOrderHeader
            ON Sales.SalesPerson.SalesPersonID = Sales.SalesOrderHeader.SalesPersonID INNER JOIN
        Sales.SalesOrderDetail
            ON Sales.SalesOrderHeader.SalesOrderID = Sales.SalesOrderDetail.SalesOrderID INNER JOIN
        Sales.SalesTerritory
            ON Sales.SalesOrderHeader.TerritoryID = Sales.SalesTerritory.TerritoryID
GROUP BY
    Sales.SalesTerritory.[Group],
    Person.Contact.FirstName,
    Person.Contact.LastName
ORDER BY
    Sales.SalesTerritory.[Group],
    Person.Contact.FirstName,
    Person.Contact.LastName
```

*2. ábra. Példaképpen egy SELECT*

Némi elemzés után azt látjuk, hogy ez bizony nagyon sok különböző kimutatás: kimutatásonként különbözőképpen paraméterezett SQL-utasításokat vagy tárolt eljárásokat kell írnunk, ráadásul más-más szempont szerint kell csoportosítani, összesíteni az adatokat.

Továbbgondolva a feladatot, arra a következtetésre jutunk, hogy egy olyan rendszer kellene, amelynek révén az adatokat a felhasználók tetszőleges szempontok szerint csoportosíthatják, összesíthetik, szűrhetik és megjeleníthetik.

Neki is állunk tehát, írunk egy olyan programot, amely lehetővé teszi, hogy a felhasználó kiválassza: milyen szempontok szerint és milyen sorrendben szeretné az adatokat csoportosítani, rendezni, és milyen szűrőfeltételeket szeretne alkalmazni. Programunk a paramétereknek megfelelően dinamikusan összerak egy SELECT utasítást, amelyet lefuttatunk az adatbázisunkon, majd az eredményből HTML alapú kimutatást vagy Excel-táblázatot készítünk. Készen vagyunk, a felhasználók elégedettek, mi pedig bezsebeljük az elismerést.

A gondok viszont csak ezután következnek. Egy év használat után a rendszerünk elkezd lassulni, a jelentések elkészítése egyre hosszabb időt vesz igénybe.

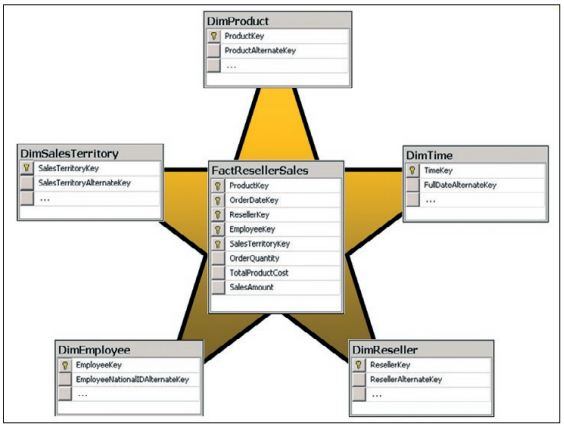
Elemezve a helyzetet rájövünk, hogy az összesítések feldolgozása nagyon megterheli az SQL Servert, ezért úgy döntünk, hogy egy másik szerverre tükrözzük az adatokat, és a kimutatásokat onnan futtatjuk.

Úgy látszik, hogy sikerrel veszzük ezt az akadályt is, azonban jön az újabb kihívás: az értékesítési adatok mellett a kereskedők kvótáját is meg kellene jelenítenünk. Ezek Excel-táblázatokban vannak ugyan, de vegyük már át őket! Elkezdjük tehát áttervezni a rendszerünket: az adatbázisunkat új táblákkal kell bővítenünk, az eddigi lekérdezéseinket meg módosítanunk kell. Azonban még be sem fejeztük a tervezést, jelentkezik az újabb igény: vegyük bele a rendszerbe az internetes kereskedelmi rendszer eladási adatait is. Na, körülbelül ez lenne az a pont, ahol komolyan azon kezdenénk gondolkodni, hogy keresünk egy másik állást. Az álláskeresés helyett azonban más megoldást is választhatunk: felépítünk egy adattárházat, amely eléggé rugalmas ahhoz, hogy különböző rendszerek adatait be tudja fogadni, és olyan a szerkezete, hogy egyszerű legyen belőle kimutatásokat vagy Excel-táblákat gyártani.

### Adattárház építése – a csillagséma

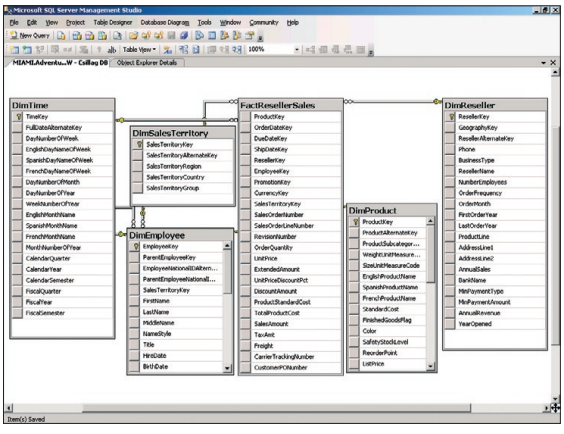
Először az adatszerkezetünket gondoljuk át. Mivel az adatbázist úgyis megtükröztük, nem okoz különösebb gondot az eredeti táblaszerkezetünk olyan átalakítása sem, hogy minél könnyebb legyen benne összesítő lekérdezéseket futtatni. Némi utánajárás, netes kutakodás alapján az adattárházak alap-

vető adatszerkezeténél, a csillagsémánál kötünk ki. A csillagsémát egy adatokat (tényeket) tartalmazó központi ténytábla, és a körülötte elhelyezkedő, csoportosítási és szűrési szempontokat tartalmazó dimenziótáblák alkotják.




*3. ábra. Csillagséma*

A ténytábla sorai adatrekordokat tartalmaznak, mégpedig a dimenziótáblákban megfelelő részletességgel: minden egyes sor egy adott termék, egy adott napon, egy adott kereskedő által, egy adott viszonteladónak, egy adott kereskedelmi területen értékesített mennyiségének, költségének és bevételének az összegét tartalmazza. Azt is mondhatjuk, hogy a kereskedő, az idő, a termék, a viszonteladó és az értékesítési terület a ténytáblában található adatok dimenziói, azaz ezek segítségével helyezhetjük el az adatokat „térben és időben", az adatok sokdimenziós terében.



*4. ábra. A csillagséma részletes szerkezete*

A központban lévő ténytábla a dimenziótáblákhoz egy-egy kulccsal kapcsolódik. A ténytábla soraiban található adatokat tehát úgy értelmezhetjük, hogy megkeressük a dimenziótáblákban a kulcsoknak megfelelő rekordokat.

A dimenziótáblák rekordjai a dimenzió tulajdonságait (más szóval attribútumait) tartalmazzák. A termékdimenzió esetében ezek a termék jellemzői: a termék neve, egységára, típusa, színe stb. Az idődimenzió attribútumai például az adott naphoz tartozó év, negyedév, hónap, nap stb.

A dimenziótáblák általában erősen redundánsak, denormalizáltak. A denormalizáció csökkenti a táblák közötti kapcsolatok számát, és alaposan leegyszerűsíti az adatszerkezetet, ezáltal csökkenti a lekérdezések elkészítésének és futtatásának az idejét.

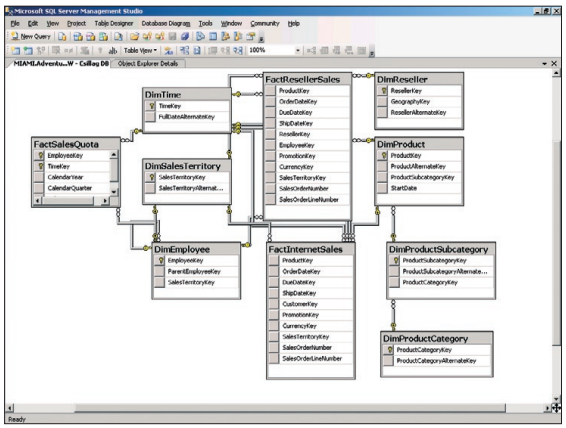
Könnyen beláthatjuk, hogy egy ilyen adatstruktúrában a lekérdezések készítése valóban lényegesen egyszerűbb, mint az eredeti modellben. Ahhoz például, hogy a területek és kereskedők alapján csoportosított kimutatást készítsünk, itt csak két táblakapcsolat szükséges, az eredeti négy helyett. Nem csak könnyebb így megírni a SELECT-et vagy az azt készítő programot, de a lekérdezés is sokkal gyorsabban tud futni, mivel kevesebb erőforrást igényel a szervertől az adatok összeszerelése.

Természetesen a csillagsémába be is kell töltenünk az adatokat. Egyszerű esetben választhatjuk azt a megoldást, hogy a csillagot, mondjuk, mindennap újratöltjük adattal. Ezt nyilván nagyon könnyű megvalósítani, de nagy adathalmaz esetén nem ez a hatékony megoldás, hanem inkább a változások átvezetése. Ennek a kivitelezése több munkát igényel ugyan, cserébe viszont könnyen lekérdezhető, jól áttekinthető adatstruktúrát nyerünk.

Az adattárházunk persze nem egyetlen csillagból fog állni, mivel egy csillagsémába nehéz lenne jól kezelhető módon minden adatot betölteni. Jó példa erre a kereskedelmi kvóták, vagy az internetes eladások esete. A kvótát kereskedőnként éves szinten határozzuk meg, így az előző csillagunk napi szintű adatai közé nem lenne szerencsés a kvótaadatot betölteni. Ráadásul a kvótához nem kapcsolódik a viszonteladó, a kereskedelmi terület és a termék. Hasonlóképpen az internetes eladások esetén szintén nincs értelme viszonteladóról beszélni.

Így az adattárházunkat három csillagra tudjuk bontani, egy-egy ténytáblát szentelve a viszonteladói és az internetes eladásoknak, és egyet a kvótának.

A dimenziótáblák viszont lehetnek közösek, azaz ugyanaz a jellemző több csillag ágán is szerepelhet. Ennek a struktúrának a kiemelkedő előnye az lesz, hogy a különböző adatok (az internetes és a viszonteladói értékesítés, valamint a kvóta) a közös dimenziótáblák mentén összekapcsolhatók lesznek.




*5. ábra. A teljes csillagséma*

Ha idáig elértünk, akkor van egy professzionális adatszerkezetünk, amire ráépíthetjük a lekérdező, elemző rendszerünket. A kérdés most már az, hogy fejlesszünk-e egy saját rendszert, vagy alkalmazzunk valamilyen létező, jól bevált megoldást. A válasz egyszerű: implementáljunk egy OLAP rendszert, és használjuk a lekérdezésekhez például Excelt vagy Reporting Servicest.

## Az OLAP rendszer implementálása

Az OLAP rendszereket pont arra találták ki, amire szükségünk van: az OLAP rendszerben az adatokat tetszőleges szempontok sze-

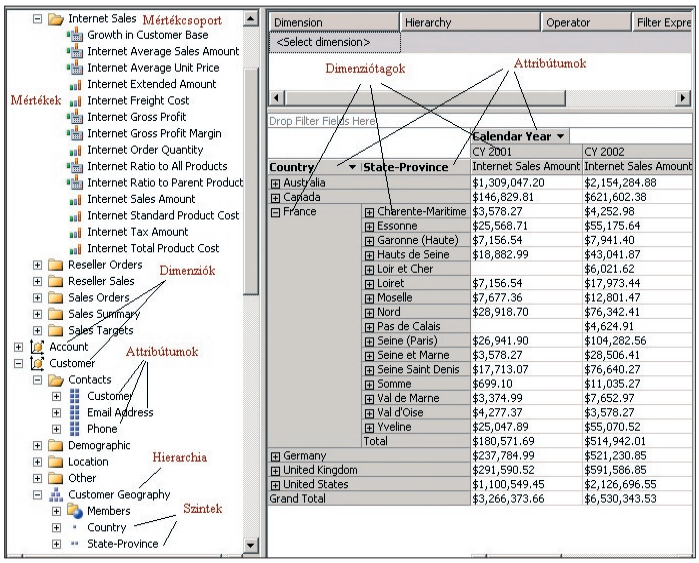
rint összesíthetjük, szűrhetjük, elemezhetjük és megjeleníthetjük. Mindezt közel valós időben, de legalábbis a relációs rendszereknél több nagyságrenddel gyorsabban.

Mivel az adataink SQL Server 2005-ben vannak, kézenfekvő a megoldás, hogy az Analysis Servicest használjuk a rendszerünk megépítésére. Az OLAP rendszer elkészítése előtt azonban tekintsük át a multidimenzionális rendszerek felépítését, funkcióit.

## Az OLAP rendszerek felépítése

Az OLAP rendszerek az adatokat dimenziómodellben kezelik. A dimenziómodellben az adatokat dimenziók mentén elemezhetjük és összesíthetjük, pontosan úgy, ahogyan a csillagséma esetén.

Az OLAP rendszerben azonban a csillagséma adatmodelljét továbbfejlesztjük: az adatokat a csillagséma alapján többdimenziós adatkockákba szervezzük, a dimenziókat pedig, az attribútumok mellett, a dimenzión belüli adatcsoportosítást és navigációt támogató hierarchiákkal is felruházzuk.




*6. ábra. Az OLAP-kocka szerkezete*

## A dimenziómodell legfontosabb elemei

**Kockák.** A kockák mértékekből és a mértékeken értelmezett dimenziókból állnak. A mértékek a csillagséma ténytáblájában található adatok, míg a dimenziókat a dimenziótáblákból hozzuk létre. Egy kocka számos különböző mértéket tartalmazhat, az egyes mértékeket pedig különböző dimenziókkal jellemezhetjük. Az adatok elemzését, lekérdezését, csoportosítását a dimenziók mentén végezhetjük el, csakúgy, mint a csillagsémában.

Az OLAP-kockát akár úgy is elképzelhetjük, mint egy többváltozós függvényt, ahol a dimenziók a változók, a függvény értékei pedig a dimenziók által meghatározott adatok. Három dimenzió esetén ez egy olyan „kocka”, amelynek élei mentén dimenziótagok (a dimenziótábla egyes attribútumainak értékei) helyezkednek el, a metszéspontjaikban található „cellákban” pedig a mértékeket találjuk.

Az OLAP-kockák adatai a dimenzió-attribútumok mentén csoportosíthatók, összesíthetők, szűrhetők, a hierarchiák mentén pedig egyszerűen navigálhatunk bennük, így alkalmasak az adatok tetszőleges csoportosításban való megjelenítésére, összesítésére, elemzésére. Egy-egy jól definiált OLAP-kocka akár jelentések százainak az elkészítésére is alkalmas lesz anélkül, hogy bonyolult lekérdezéseket kellene rajta definiálnunk.

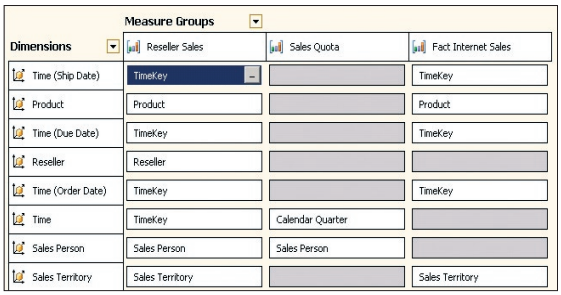
Az OLAP-kockákat a csillagséma alapján készítjük el: a mértékeket a ténytáblák, a dimenziók adatait pedig a dimenziótáblák tartalmazzák.

Az eddigiek alapján könnyen felmerülhet a kérdés: ha egyszer az OLAP-kocka a csillagsémára épül, miért van rá szükség? Miért nem a csillagsémát használjuk az adatok lekérdezésére?

A kérdésre a válasz nem triviális, de röviden összefoglalható: az OLAP-kockákban előre, a kockák adatainak betöltését követően összesíthetjük és indexelhetjük az adatokat, így nagyságrendekkel csökkenthetjük a lekérdezési időt. Emellett az OLAP-modellek az adataink hierarchikus szerkezetét is tartalmazzák, amely megkönnyíti az összesítések és a lekérdezések definiálását, meggyorsítva ezek végrehajtását.

**Dimenzió-hierarchiák.** Az OLAP-modellek fontos elemét képezik a dimenzió-hierarchiák. A dimenzió-attribútumok gyakran rendezhetők olyan hierarchikus kategóriákba, mint például „év – negyedév – hónap – nap” vagy „ország – megye – város”. Ezeket az adatokat a csillagséma dimenziótáblái tartalmazzák, de nem mondanak semmit az egymáshoz fűződő kapcsolatukról. A hierarchiák egymásra épülő szintekből állnak, a szintek felülről lefelé haladva (például ország – megye – város) egyre részletesebben jellemzik az adatokat. A hierarchiák mentén összesíthetjük az adatokat, vagy navigálhatunk bennük: „lefúrhatunk” például az év szintjéről a negyedév szintjére, hogy lássuk az éves összesített adatok negyedéves részletezését, vagy „felfúrhatunk” egy város szintű adatról országos szintűre, így kiszámíthatjuk például egy város teljesítményét az ország teljesítményéhez viszonyítva. A dimenzió-hierarchiák általában tartalmaznak egy legfelső „összesen” szintet, amely a hierarchia felső szintjéhez tartozó adatok összesítését támogatja.

Dimenzió-hierarchiának tekinthetünk egyetlen dimenzió-attribútumot is, ebben az esetben a hierarchiának két szintje van, az egyik maga az attribútum, a másik pedig az „összesen” szint. Az egy attribútumból álló hierarchiákat attribútum-hierarchiáknak, míg a több attribútumból állókat felhasználói hierarchiáknak hívjuk.




*7. ábra. Mértékcsoportok és dimenziók kapcsolata*

**Mértékcsoportok.** Az OLAP-kockában – csakúgy, mint a csillagsémában – nem minden mértékre értelmezhető minden dimenzió, és az is gyakran előfordul, hogy az adatokat nem ugyanolyan részletességgel gyűjtjük.

A kereskedők kvótáját nem bontjuk meg termékek szerint, van viszont negyedéves és területi bontás. Ugyanakkor az eladási adatainkat termékek szerint is csoportosítjuk, és napi szinten tároljuk. A mértékeket, a rájuk értelmezett dimenzióknak és a részletezettségüknek megfelelően mértékcsoportokba soroljuk: azok a mértékek kerülnek egy csoportba, amelyekre ugyanazok a dimenziók vonatkoznak, és ugyanolyan részletezettségűek. Az összetett csillagséma minden egyes csillagának egy mértékcsoport felel meg a kockában. A mértékcsoportokhoz más-más ténytábla tartozik, a dimenziótáblákat pedig eltérő módon kapcsolhatjuk azokhoz. A különböző mértékcsoportok adatait a közös dimenziók mentén együttesen is elemezhetjük.

## Az UDM

Az Analysis Services 2005 dimenziómodellje az UDM, azaz az egyesített dimenziómodell (angolul Unified Dimensional Model). Az UDM a hagyományos dimenziómodellek előnyös tulajdonságait további fontos elemekkel egészíti ki, amelyek az Analysis Servicest alkalmassá teszik a relációs és a multidimenzionális jellegű adatstruktúrák hatékony modellezésére is. Az UDM a flexibilis adatmodellezésen kívül számos olyan funkciót is tartalmaz, amely a dimenziómodellt gazdagabbá, jobban felhasználhatóvá teszi.

Az UDM a dimenziómodellezés mellett lehetővé teszi, hogy a kockáinkban számításokat végezzünk. A számítások nyelve az MDX (Multi Dimensional Expressions), amelynek speciális függvényeivel és kifejezéseivel az OLAP-kockákban navigálhatunk, halmazokat képezhetünk és számításokat végezhetünk.

Az UDM támogatja a KPI-k készítését és lekérdezését; a modellek több nyelvre fordítását (Translations); a dimenziókhoz, egyes adatokhoz kapcsolt akciók (például jelentés megnyitása, részletek megjelenítése) definiálását és végrehajtását; a kockák nézeteinek definiálását (Perspective); az adatbázisok particionálását; az adatok különböző adatstruktúrákban történő tárolását (Multidimenzionális OLAP, Relációs OLAP, Hibrid OLAP).

A felsoroltakon kívül még nagyon sok kényelmi, teljesítmény- és megbízhatóságnövelő tulajdonsággal rendelkezik.

**UDM-ügyfelek.** Az UDM-hez különböző interfészeken keresztül sokféle eszközzel férhetünk hozzá.

Az UDM-ben létrehozott kockáinkat használhatjuk az Excelből, a Visióból, a Reporting Servicesből, a Business Scorecard Managerből, a SharePointból, a PerformancePoint Serverből és a ProClarity-ből, de más gyártók elemző, adatmegjelenítő alkalmazásai is támogatják, mint például a Cognos vagy a Business Objects.

A felhasználók így nemcsak az általunk fejlesztett lekérdezőalkalmazást használhatják, hanem a legkülönbözőbb rendszerek között válogathatnak, az igényeiknek (na meg a pénztárcájuknak) megfelelően.

**Fejlesztés, telepítés, karbantartás.** Az UDM-modellek fejlesztésére, telepítésére és karbantartására számos eszköz áll rendelkezésünkre: általában a Business Intelligence Development Studióval (Visual Studio 2005) és az SQL Server Management Studióval interaktív módon, míg az XMLA-szkriptek, AMO (Analysis Management Objects) használatával programozottan végezhetjük el a munkát.

A feladatok végrehajtását az SQL Server Integration Services és az SQL Server Agent segítségével automatizálhatjuk.

Country-Region: France

| Sales Amount / Category | Subcategory | Product Name | Color | Calendar Year CY 2004 | CY 2003 | Grand Total |
|---|---|---|---|---|---|---|
| Accessories | | | | 198,943 | 322,390 | 521,333 |
| Bikes | | | | 14,256,061 | 26,529,771 | 40,785,832 |
| Clothing | | | | 400,427 | 884,245 | 1,284,671 |
| | Bib-Shorts | | | | 64,877 | 64,877 |
| | Caps | | | 6,907 | 14,682 | 21,588 |
| | | AWC Logo Cap | Multi | | 4,705 | 4,705 |
| | | AWC Logo Cap | Multi | 6,907 | 9,977 | 16,883 |
| | Gloves | | | 18,146 | 103,258 | 121,404 |
| | Jerseys | | | 159,803 | 296,723 | 456,526 |
| | Shorts | | | 114 | 11 | 296,581 |
| | Socks | | | 7 | 93 | 18,353 |
| | Tights | | | | 37 | 78,937 |
| | Vests | | | 93,142 | 133,263 | 226,405 |
| Components | | | | 2,091,012 | 5,482,497 | 7,573,509 |
| Grand Total | | | | 16,946,442 | 33,218,903 | 50,165,345 |

Sales Amount
Value: 159,803
Row: Clothing - Jerseys
Column: CY 2004

*8. ábra. OLAP-kocka az Excelben*

## A megoldás

A fentiek alapján a megoldásunk egy három mértékcsoportot (Reseller Sales, Internet Sales, Sales Quota) tartalmazó UDM-kocka lesz, amelynek adatait öt dimenzió (Product, Sales Territory, Reseller, Time, Employee) mentén elemezhetjük. A kockák alapjául nyolc relációs tábla szolgál: három ténytábla (FactResellerSales, FactInternetSales, FactSalesQuota) és öt dimenziótábla (DimProduct, DimSalesTerritory, DimReseller, DimTime, DimEmployee).

A felhasználók az adatokat Reporting Services-zel készített jelentésekben kérdezhetik le, vagy közvetlenül az Excelben elemezhetik.

Példaképpen lássuk, mire képes az UDM az Excelben. Az Excel-felhasználók az UDM-ből közvetlenül kimutatásokat és diagramokat készíthetnek, a megjeleníteni tervezett dimenziókat és mértékeket tetszőlegesen megválasztva.

Természetesen az adatokat feltételek alapján szűrhetik, és tovább is csoportosíthatják újabb összesítéseket képezve. A megjelenítést feltételes formázással egészíthetik ki, így például kiemelhetik az átlagosnál jóval gyengébben eladott termékeket. A kimutatások használatához nem kell speciális ismeret, elegendő az OLAP-kockák dimenzióit, mértékeit és az Excel-kimutatás (pivot) működését megérteni.

Megoldásunk tehát látványos, gyors, skálázható és végül, de egyáltalán nem utolsósorban bővíthető: az újabb adatok elhelyezhetők a meglévő ténytáblákban, vagy újabb ténytáblákat és dimenziótáblákat illeszthetünk be, a kockánkat pedig ennek megfelelően alakíthatjuk át.

Egy ilyen megoldás implementálása egy-két héttől néhány hónapig terjedhet, attól függően, hogy milyen adatokkal dolgozunk, és milyen egyéb követelményeknek kell megfelelnünk.

## További információk

Annak, aki szeretne megismerkedni az Analysis Services által nyújtott szolgáltatásokkal, lehetőségekkel és az UDM-modellek építésének részleteivel, érdemes elolvasnia az SQL Server Analysis Services 2005 Step by Step (Microsoft Press – http://www.microsoft.com/mspress) könyvet vagy Teo Lachev: Applied Analysis Services 2005 (Prologika Press – http://www.prologika.com) című könyvét. Természetesen OLAP-tanfolyamok a hivatalos Microsoft-oktatóközpontoknál is elérhetők.

*Kovács Zoltán*
vezető oktató
(kovacsz@szamalk.hu)
Számalk

# Saját Microsoft

## „Lehozzuk a technológiát földközelbe” – interjú Kőnig Tiborral.*

**T. M.: A szakmai közösség legtöbb tagjának aligha kell bemutatkoznod, az „újak” számára viszont mondd el: ki vagy és mivel foglalkozol a Microsoftnál!**

K. T.: Kőnig Tibor vagyok, szakmai vezető a nagyvállalati csoportban. Szakmai vezetőként az a feladatom, hogy az ügyfelek komplexebb projektjei kapcsán – amelyek különféle technológiákat érintenek – fogjam össze és koordináljam a megfelelő szakértőinket. Szintén hozzám tartozik a szakmai napok rendezése, az itt elhangzó előadások felépítésének, mélységének és tartalmának kitalálása.

*Kőnig Tibor*

Emellett rendszermérnökként is működöm, azon belül is a SharePoint egyes elemeivel foglalkozom: űrlapkezeléssel, üzletiintelligencia-szolgáltatásokkal, valamint a nagyvállalati projektkezeléssel, tehát a Project Serverrel, Project Professionallel. Sajnos egyre csökkenő mértékben, de még a TechNet szemináriumokon és más külső eseményeken is részt veszek.

**T. M.: Hogyan tudsz lépést tartani ilyen sok technológiával?**

K. T.: Nagyon régóta itt vagyok már a cégnél, gyakorlatilag én voltam az első technikai ember. Ezért tulajdonképpen együtt nőttem a termékportfólióval. Igazából itt arra vigyáztam, hogy valamilyen szintű átlátóképességet megőrizzek: legyen egy olyan keretrendszer a fejemben, amelyben el tudom helyezni a mi termékeinket, legyen gyakorlatilag bármilyen üzleti területről szó, és persze ennek a keretrendszernek elég rugalmasnak kell lennie, hogy a partnereink is beleférjenek, sőt adott esetben a versenytársaink is. Sajnos, mondjuk, a Zune vagy az Xbox már kívül esik ezen a körön, bár többen szoktuk mondogatni, hogy itt az álomállás valamelyik játék termékmenedzseréé lenne.

**T. M.: Mióta foglalkozol Microsoft-technológiákkal? Mikor kerültél ide?**

K. T: ’93 decemberében, egy karácsonyi partyn álltam először a Microsoft színeiben az ügyfelek rohamát – az Operaházban volt –, de igazából úgy kerültem kapcsolatba a Microsofttal, hogy ’92-ben, amikor az Olivetti nevű cégnél dolgoztam, elküldtek egy 11 hetes tréningre. Az Olivetti egy hagyományos értékeket nagyon szem előtt tartó vállalat volt – jól tönkre is ment már azóta –, és ennek a 11 hétnek az első öt hete szakmai tréning volt. Hagyományos unixos megközelítéssel, aminek a végére mi imádkoztuk oda a Novellt, hogy legalább egy rendes technológiáról hadd halljunk többet.

A legutolsó napján ennek a tréningnek elvittek minket egy olyan terembe, ahol addig még nem jártunk – ott volt 5 PC-nek látszó tárgy, de valójában gyorsan kiderült róluk, hogy sokkal erősebb munkaállomások, ráadásul nem is inteles, hanem RISC-processzorokkal, és egy olyan operációs rendszer futott rajtuk, amire azt mondták, hogy az a Windows NT. ’92-ben még nem volt kész a szoftver teljesen. Ezek a gépek – máig emlékszem rájuk – több tartományban futottak, mindegyik gép egy-egy tartományvezérlő volt. Egy picit beszéltek erről nekünk, és nagyon felkeltették az érdeklődésemet, hogy a Microsofttal foglalkozzam.

Korábban egy angol cégnél szoftverfejlesztő voltam, tőzsdei alkalmazásokat készítettünk, amelyeknek a kliensoldala Windows 3.0-n futott. Tehát azt már tudtam, hogy a Windowsban nagyon nagy üzleti potenciál van, mindenki Windowst akart használni a felülete miatt. Viszont annak a 3.0-s verziónak a korlátaival is pontosan tisztában voltam, és akkor a tréningen azt láttam, hogy itt van egy új operációs rendszer, amivel ez átléphető. Úgyhogy alig több mint egy év múlva átnyergeltem a Microsofthoz.

**T. M.: Az elmúlt több mint 13 évben melyek voltak a legemlékezetesebb élményeid a Microsoftnál?**

K. T.: A redmondi látogatások. Azt gondolom, hogy ez bárkinek nagy élmény, aki az informatikában benne van, és nem valamiféle vallásos utálattal viseltetik a Microsoft iránt. Nekünk, a Microsoftnál dolgozóknak ez az egyik Mekkája az informatikának. Én mindig nagyon szívesen megyek oda, olyan, mint egy városrész – valójában nem felhőkarcoló székházakat kell elképzelni, hanem

* Rövidített, szerkesztett változat, az interjú teljes szövege és hangfelvétele megtalálható a következő URL-en: http://tinyurl.com/38fr7w

egy sokkal nyugodtabb, békés környezetet, ahol tavak, focipályák, egyemeletes épületek vannak, amelyek között csak úgy mászkálni is kellemes.

Elmondanék még egy személyes élményt is: az első komoly feladat amit a Microsoftnál kaptam, egy termékbejelentés megszervezése és lebonyolítása volt. Méghozzá a Windows for Workgroups 3.11-es, illetve a Word 6.0 változatának bejelentése. Helyszínül a Planetáriumot választottuk. Akkoriban nagy divat volt a különböző előadók zenéjére készített lézershow-knak a bemutatása, ezért egy egyedi lézershow-t készíttettünk mi is azzal a céggel, amely ezt egyébként a Planetárium számára végzi. Nagyon érdekes Microsoft-képeket használtak fel hozzá – emberi kézből egér lett, és hasonló lézeres áttűnéseket láthattunk, de az az igazság, hogy kicsit megcsúsztak időben, és nem készültek el elég hosszú anyaggal. Én kértem tőlük – már nem emlékszem a pontos időkre, de talán – 15 percet, és 8 percnyi anyaggal készültek el mindössze. De megnyugtattak: azt mondták, hogy van egy témájában gyakorlatilag teljesen idevágó saját készítésű anyaguk, amit majd hozzátoldanak ehhez, és úgy rendben lesz minden.

Nem volt időm megnézni a kész anyagot, viszont a termékbejelentésre eljött az akkori európai elnök. Az eleje jól indult, az összes informatikai motívum bejött a közönségnek is. Aztán egyszer csak azt láttuk, hogy a Microsoft-jelképek afrikai halotti maszkokká változnak, és mindenféle vadászó bennszülöttek képei jelennek meg; a zene is ennek megfelelően változott meg. Döbbenten figyelt mindenki. Én ott, miközben elsüllyedtem a székben, azt gondoltam, hogy most már holnap be sem kell jönnöm dolgozni. De ha nem is rúgnak ki, énrám többé pénzt ennél a cégnél nem bíznak. Utána magához kéretett az európai elnök, és azt mondta, hogy ez egy rendkívül érdekes előadás volt, bár ő nem teljesen értette a kontextust, amelyben ezek az afrikai dolgok előkerültek, de úgy látta, hogy a közönségnek nagyon tetszett. Szerinte egy kicsit jobban kéne fókuszálni, de maga a bemutató, az remekül sikerült – úgyhogy gratulál.

Megdöbbentő élmény volt! Én tényleg azt hittem, hogy itt vége a microsoftos karrieremnek, és kiderült, hogy nem, ennél a cégnél lehet hibázni, az a lényeg, hogy ne sokszor, illetve látszódjon az az irány, amibe te akartál menni, és tudj tanulni a saját hibáidból.

**T. M.: Szemben ezzel a kis bakival, mire vagy igazán büszke az elmúlt pár évből?**

K. T.: Egyértelműen büszke vagyok a TechNet sorozatra, ott is elsősorban azt emelném ki, hogy nagyon sok microsoftos és nem microsoftos előadó van, akiket szívesen fogad a közönség, akik felnőttek, és nagyon megtanultak előadni, nagyon megtanulták értelmesen átadni azt a sok információt, amelyeket ezekkel a termékekkel kapcsolatban át is kell adni. Lojális közönségünk van, és azt gondolom, hogy senki másnak nincs az országban ilyen jellegű tudástranszfer-eseménye. Amikor idejöttem a Microsofthoz, az eredeti cél az volt számomra, hogy olyan cégnél dolgozzam, amelyik a magyar piacon is széles körben használt szoftvereket tud előállítani, és amelyik képes arra, hogy a magyar informatikai kultúrát magasabb szintre emelje. Azt gondolom, hogy a TechNet egyértelműen nagy lépés ebbe az irányba.

**T. M.: Mi az, ami még ennyi idő után is itt tart a cégnél?**

K.T.: Az az érdekes a Microsoftban, hogy szépen, lépésről lépésre megtalálja azokat a felhasználási területeket, amelyek korábban csak fehérköpenyes, nagyon szakértő, nagyon sok pénzért dolgozó cégek territóriumának számítottak. Ezeket lehozza a földközelbe, mind az árukat, mind a szolgáltatásaik elérhetőségét tekintve. Olyan terméket csinál belőlük, amely a cégek sokkal nagyobb része számára elérhető, honosítja őket. Ezzel pedig olyan szerepet tölt be, amilyen nincs még egy az informatikában.

A rendszermérnökök szerepe pedig jól rímel erre. A mai magyar nagyvállalatoknál – mint mindenhol máshol a világban – már igen sok technológia van. Ezeknek az értelmes áttekintése, annak a kitalálása, hogy ha már egy adott szinten állunk, hogyan lépjünk tovább; annak a felfedezése, hogy milyen komponensekre van ahhoz szükségünk, hogy pluszszolgáltatásokat tudjunk nyújtani a saját ügyfeleinknek, felhasználóinknak, nem egyszerű feladat.

A rendszermérnökök pont abban segítenek, hogy a szükséges eszközök közötti választást megkönnyítsék, lehetővé tegyék. Amit a Microsoft csinál nagyban, mi is azt csináljuk kicsiben. Nagy élmény ezzel foglalkozni, és folyamatosan kihívást nyújt.

**T. M.: Mennyire tudod elválasztani a munkát a szabadidőtől?**

K. T.: Amikor csatlakoztam a Microsofthoz, majdnem minden hétvégén dolgoztam. Nagyon sok munka volt, és egyedül voltam. Ahogy a rendszermérnöki csapatot építgettem, lassan összeállt egyfajta munkamegosztás, amelyben szorosan együttműködünk a tervezést és bevezetést végző MCS-sel, vagyis a Microsoft nagyvállalati szolgáltatások üzletágának konzulenseivel, valamint a szakmai ügyfélszolgálattal is.

A terhek folyamatosan csökkentek, így már nem feltétlenül kellett dolgoznom a hétvégeken, de ez nagyjából egybeesett azzal az időszakkal, amikor a notebookok elérhetővé váltak. Ezen kívül a Microsoft fizette az akkor még telefonos betárcsázási díjat, magyarán az otthonról való munkavégzés lehetségessé, sőt adott esetben kívánatossá vált.

Ma már tulajdonképpen azt lehet mondani, hogy a munka kötelező részét el lehet végezni napi 8 órában is. Azonban nekem a munkám legalább annyira a hobbim, mint a tényleges pénzkereső foglalkozásom, úgyhogy még így is nagyon sok időt fordítok rá. Egyetlenegy megszorítást tettem, amikor a második gyerekem is megszületett, és már óvodás korú lett: akkor megállapodtam az akkori főnökömmel – és ezt azóta is frissítettem minden további főnökömre –, hogy hetente két napon négy órakor elmegyek a cégtől, és akkor én megyek értük az iskolába.

**T. M.: Ez tényleg elvárt, hogy ilyen sokat dolgozzon mindenki?**

K. T.: Most már nagyon sokan vagyunk a leányvállalatnál, de amikor én jelentkeztem, és utána is még jó sok évig az volt a jellemző, hogy nem olyan embereket keresett a cég, akik a főnökük által kiszabott munkát fogják végrehajtani, hanem akik átlátják a saját munkakörükhöz szükséges feladatokat, maguknak tűzik ki, hogy mit szeretnének csinálni, és ezt meg is valósítják.

Azt szoktuk erre mondani, hogy itt a cégnél mindenkinek megvan a saját Microsoftja, és addig csinálja ezt, amíg úgy gondolja, hogy legalább annyit kap a cégtől, illetve a céggel való foglalkozástól, mint amennyit ő maga beletesz.

# OFFICE 2007 OKTATÁSOK A NETACADEMIÁNÁL!

A NetAcademia 2007-ben gyakorlatias, a napi feladatokra koncentráló Office 2007 képzéseket indít végfelhasználók és rendszergazdák számára!

**Kiknek ajánljuk a képzéseket?**

- Infómunkások, irodai alkalmazottak
- Adatrögzítők, adatelemzők
- Sales- és marketingterületen dolgozók
- Vezetők és asszisztensek

**Jelentkezzen most és használja ki akciónkat!**

OFFICE 2007
A NETACADEMIA OSZTOTT.
HÚZZON EGY ÜTŐS LAPOT!

## A NetAcademia Office 2007 tanfolyamkínálata

| | |
|---|---|
| Office 2003-2007 áttérés | 5 nap |
| Word 2007 alapozó | 2 nap |
| Word 2007 haladó | 3 nap |
| PowerPoint 2007 | 3 nap |
| Excel 2007 alapozó | 2 nap |
| Excel 2007 haladó | 3 nap |
| Access 2007 alapozó | 2 nap |
| Access 2007 haladó | 3 nap |

Az Office 2007-es képzések mellett az Office 2003 tanfolyamok is megtalálhatók kínálatunkban.

*További információk az oktatásokról:*
WWW.NETACADEMIA.NET/OFFICE

## Nyári bevezető Office akció*

Több Office tanfolyam együttes megrendelése esetén óriási kedvezményt kínálunk:

- Két Office tanfolyam után 10%
- Három Office tanfolyam után 15%
- Háromnál több Office tanfolyam után 20%

Minden Office tanfolyam megrendelés részt vesz egy sorsoláson, ahol a fődíj egy dobozos Windows Vista Ultimate!**

**Az akciós időszak 2007.06.01-2007.09.30. között tart.*
*** A sorsolás időpontja 2007.10.01.*
*Csak az online megrendelések között sorsolunk!*

netACADEMIA
A LEGJOBBAKAT TANÍTJUK.


Cím: 1062 Budapest, Andrássy út 62.

Telefon: (06 1) 472-1214
Irodai mobil: (06 20) 369-6947
Fax: (06 1) 472-1215

Internet: www.netacademia.net
E-mail: info@netacademia.net

# MICROSOFT DYNAMICS

**A Microsoft Dynamics rendszerek a vállalatirányítás következő generációját képviselik.**

**Értéket adunk vállalatának:**
Iparági és speciális megoldások,
az autókereskedelemtől az építőiparig, a kiskereskedelemtől az élelmiszeriparig,
melyet több mint 40 hazai, az adott iparágban jártas partnerünk forgalmaz és támogat.

**Értéket adunk munkatársainak:**
A rendszer könnyen használható, mert a Microsoft Office rendszerek mintájára kialakított kezelőfelületek már az első pillantásra is ismerősek, gyorsan átláthatóak.

Ismerje meg: **www.microsoft.hu/dynamics**

Microsoft Dynamics