

1.344 FT

II.  
ÉVFOLYAM  
4. SZÁM

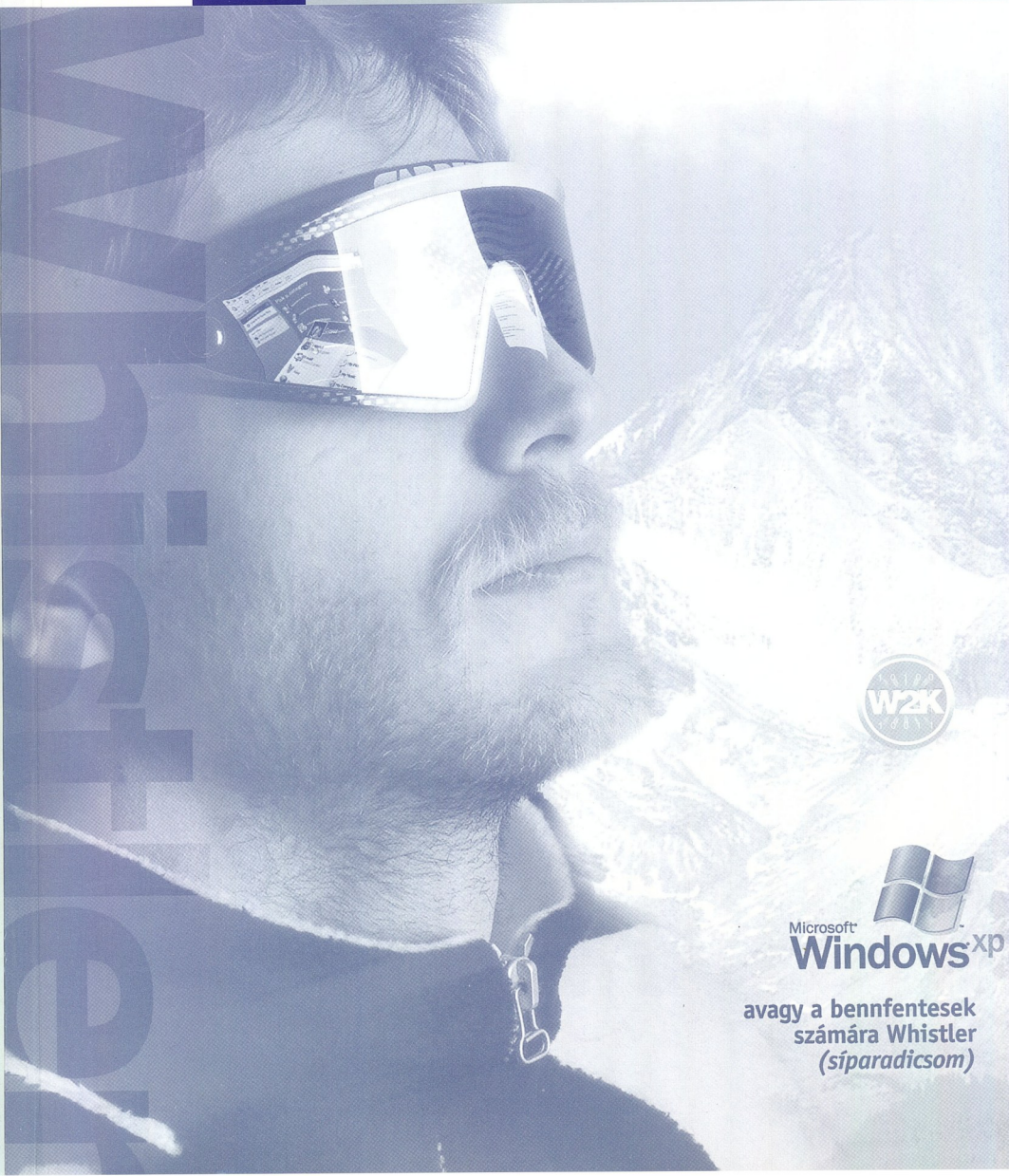
ISSN 15065105



04

# tech.net

A MICROSOFT MAGYARORSZÁG SZAKMAI MAGAZINJA



avagy a bennfentesek  
számára Whistler  
(síparadicsom)



2000. október 12–én jelentette be San Francisco-ban, az Intel Exchange konferencián Bill Gates a .NET rendszerfelügyeleti stratégiát, melyről a .NET kiszolgálók körüli sürgős-forgás közepette hazánkban eddig kevesebb szó esett. Pedig nagy változások történtek ezen a téren, illetve vannak még készülőkifeben.

Rögtön először itt van maga a tény, hogy egy-egy rendszerfelügyeleti stratégiát hirdettek meg a Microsoft. A cégen belül egy több mint

500 főt számláló fejlesztőréteg jött létre, amely kizárólag menedzsment technológiákkal foglalkozik. A részleg a Windows 2000 fejlesztőcsapat alá került, így közvetlenül Brian Valentine a vezetője (ő korábban az Exchange fejlesztését vezette, egészen az 5.5-ös verzióig).

Most tehát Brian Valentine alá tartozik a Systems Management Server fejlesztése is, ő pedig (a korábbi termékein is megfigyelhetjük) igazán ad a kód minőségére. Az SMS továbbra is stratégiai fontosságú menedzsment terméke a Microsoft-nak – már megjelent az SMS 2.0 SP3, a következő mérföldkő pedig az SMS „Topaz” kódnevű változata lesz, mely valamikor idén téle várható. A „Topaz” nem lesz „SMS2000” csak „2.x”, tehát csak egy kisebb verzióváltás. Főleg a minőségre összpontosít, az új funkciók az Active Directory integráció és a mobil ügyfelek jobb támogatása lesznek.

Fontos látnunk az SMS szerepét a termékstratégiában, hiszen sajnos sok helyen már az új, Microsoft Operations Manager terméket emlegetik az SMS utódjaként. Erről szó sincs! (Sajnos a Tech.net magazin egyik korábbi számában is hibásan jelent ez meg). A Microsoft Operations Manager (már beceneve is van: MOM) egy teljesen más területre ad megoldást: míg az SMS az új hardverelemek, operációs rendszerek és alkalmazások bevezetését, tehát a környezet megváltozását kezeli a leltározási és szoftverdisztribúciós funkciókkal, addig a MOM már ennek a megváltozott környezethez az üzemeltetésére szolgál. (Mert amit bevezettünk, azt ugye még folyamatosan figyelni is kell, hogy jól működik-...) A MOM az SMS-beli Health Monitor feladatát valósítja meg nagyságrendekkel magasabb szinten: esemény-, és teljesítményadatok gyűjtését és konszolidálását több forrásból, SQL Server adatbázisba (WMI, SNMP csapdák, Windows NT/2000 rendszernapló, alkalmazásnaplók, stb.), automatikus válaszlépések és riasztások megtételét bizonyos események bekövetkezésekor illetve teljesítményküszöbök átlépésekor, és jelentések készítését az összegyűjtött adatokból hibakeresés és kapacitástervezés céljából.

A MOM egyelőre csak a Microsoft alkalmazáskiszolgálóinak üzemeltetését fogja támogatni, azonban a lényege éppen a fejlesztethezősége van: a MOM egy üzemeltelési platform, melynek nyílt az SDK-ja, bárki fejleszthet hozzá modulokat, ún. „Management Pack”-eket. Az első ilyen cég a NetIQ lesz, nem véletlenül: a MOM eredeti kódját ugyanis töltik licenclte a Microsoft. A NetIQ a MOM megjelenése után nem folytatja saját Operations Manager termékének gyártását, helyette a MOM-hoz illeszkedő, harmadik gyártók termékeinek üzemeltetésére szolgáló csomagokat készít majd. (SAP, Notes, Oracle, ArcServer, Compaq, HP, Dell, IBM hardverek moduljai, stb.) A .NET rendszerfelügyeleti stratégiának éppen ez a központi

eleme: az első célkitűzés, hogy a Windows környezet jól kezelhető platform legyen, a második az, hogy maguk a menedzsmenteszközök fussanak Windowson, és akár innen felügyeljenek más platformot is. Ennek titka természetesen az operációs rendszerbe épített gazdag felügyeleti technológiákban és szolgáltatásokban rejlik, melyek megkönnyítik a fejlesztők dolgát. Nincs messze az idő, amikor a NetWare, Linux, Solaris kiszolgálóinkat is Windows platformról menedzseljük (MOM menedzsment csomag legalábbis lesz ezekhez a platformokhoz).

Idővel tehát a legutóbbi rendszerfelügyeleti funkció beleolvad majd az operációs rendszerbe. Erre jó példa a még csak Béta 2 állapotban lévő Windows XP Professional: itt az asztali operációs rendszernek is része a terminálszolgáltatás, így a társírányítási feladatok beépítetten, minden további eszköz telepítése nélkül oldhatók meg.

Persze azt elfelejtettem említeni, hogy minden XML alapú lesz. Az üzleti folyamatokat majd a BizTalk képi le üzemeltetési és rendszerfelügyeleti lépésekre, és vezényli XML alapon az egyes eszközök működését. Az Interneten át könnyedén kommunikálhatnak a menedzsmenteszközök HTTP alapon, XML-be ágyazott SOAP üzenetekkel. Mivel bárhol, bármikor, tetszőleges eszközzel tudunk kell menedzselni a rendszerünket, az MMC alapú konzolok szerepét a webes konzol, az MMC Portál váltja fel. Ez a felszínen ugyan csak egy HTML megjelenítő réteg, de a mélyen XML-en alapuló, minden egyes művelet szkriptelését megengedő keretrendszer lapul.

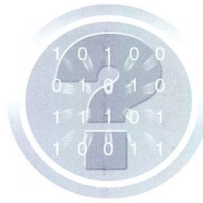
Az ilyen környezetek felügyeletéhez és üzemeltetéséhez persze sokéves szakmai tapasztalat szükséges. Ezt a gyakorlati tudást és tapasztalatot próbálja meg rendszerezetten átadni a Microsoft az Operations Framework üzemeltetési keretrendszerben. A MOF nem egy termék, hanem egy módszertan: fizikailag az üzemeltetés és egyes területein bevált fogások, megoldások, szabályok és ajánlások dokumentációinak gyűjteménye. Az üzemeltetésnek egyébként része a terméktámogatás (Help Desk) is, ahol a Microsoft-nak egyelőre nincs teljes terméke, ám a Windows XP Professional-ban már megjelenik a Support Automation Framework (SAF), mely egy fejlesztési keretrendszer – eszközkészlet, eljáráskönyvtár ad a támogatási rendszerek fejlesztőinek az alkalmazások készítéséhez. Talán mondani sem kell, hogy a SAF is XML alapú, így harmadik gyártók Help Desk eszközei könnyen integrálhatók vele. A Windows XP így valóban jól felügyelhető, üzemeltethető és támogatható vállalati asztali ügyfél lesz.

Úgy tűnik tehát, hogy a 2001-es év a rendszerfelügyelet és az üzemeltetés jegyében (is) telik a Microsoft-nál. Május 2-án a LZMét házban azonban mindenképp. A TechNet szeminárium-sorozat keretében Tarsoly Balázs kollégámmal ekkor a fenti technológiákról egy kicsit bővebben is beszélünk. Mindenkit szeretettel várunk.

Horváth Tamás  
Rendszermérnök  
Microsoft Magyarországi  
tamas@microsoft.com



# 2001 Április



**tech.net**

A Microsoft Magyarország Szakmai Magazinja

Szerkesztőség

Főszerkesztő: **Fóti Marcell**  
 marcellf@netacademia.net

Főszerkesztő-helyettes: **Fülöp Miklós**

mick@netacademia.net

Szerkesztőség címe:

1105 Budapest, Ihász utca 13.

Tel.: 263-2732

technet@netacademia.net

Nyilvános levelezési lista:

tech.net@lyris.netacademia.net

Kiadja és terjeszti  
 a **NetAcademia Kft.**

Terjesztési, előfizetési információ:

Tel.: 263-2732

terjesztes@netacademia.net

Megjelenik havonta, ára 1.344 Ft

Előfizethető megrendelőlevélben a  
 szerkesztőségénél:

1105 Budapest, Ihász utca 13.

Fax: 261-7145

<http://technet.netacademia.net/subs>

Hirdetésfelvétel:

**Bársonykalapács Marketing**

Felelős: **Udvarov Rita**

Tel./Fax: 214-0923

velvethammer@ahol.com

1027 Budapest, Fő utca 67. V. 1.

Grafikai tervezés, kivitelezés,

nyomdai előkészítés:

**Bársonykalapács Marketing**

Művészeti vezető: **Balogh Zoltán**

Nyomda:

**Cerberus Kft.**

1066 Budapest

Lovag u. 14.

Felelős vezető: **Schmidt Gábor**

ISSN 1586-5185



Hírek ..... 3. old.

## Windows 2000

Whistler ..... 5. old.

Network Monitor (V. rész) ..... 8. old.

A kernelarchitektúra fejlődése ..... 12. old.

Mixed vs. Native ..... 17. old.

Lemezkvóta a gyakorlatban ..... 19. old.

Szerszámosláda (I. rész) ..... 24. old.

## BackOffice

SMTP az Exchange 2000-ben ..... 27. old.

## Szabványok

Mime (I. rész) ..... 33. old.

## Jogi esetek

Elektronikus kereskedelem ..... 35. old.

## Developer

ASP suli (IV. rész) ..... 38. old.

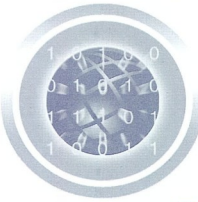
## Dupla KV

..... 43. old.

## Bill Gates mondja

Bárki, bármikor, bárhol ..... 45. old.

# Internet Explorer 6



## Nyilvános és letölthető az Internet Explorer 6 béta

Az Internet Explorer 6 preview webhelyéről [1] letölthető az Internet Explorer 6 béta változata. Akik már látták, azt mondják, hogy az IE6 nem jelent majd óriási lépést a jelenlegi verziókhöz képest. Bár levelezésért és a hírolvasásért felelős komponens, az Outlook Express egyáltalán nem változott lényegesen, a böngészőben azért lesz néhány érdekes újdonság:

- Kép-eszközár (*Image Toolbar*): néhány új funkció a képek kezeléséhez, kép mentése, küldése, nyomtatása közvetlenül az oldalból
- Automatikus képátméretezés: ha egy kép kilógna a böngésző felületéről, az IE6 automatikusan átméretezi azt (*a méretezés követi a böngésző ablakának méretváltozásait is*)
- Beépített üzenetkezelés: az MSN Messenger azonnali üzenetkezelő program beépült a böngészőbe, ezentúl könnyen elérjük a címlistánkban található felhasználókat és az instant messaging szolgáltatásokat
- P3P és egyéb biztonsági szolgáltatások: a P3P a World Wide Web Consortium (*W3C*) által fejlesztett szabvány [2]. A P3P-nek köszönhetően a felhasználók közvetlenül meghatározhatják, hogy egy adott webhely milyen információkhoz fér hozzá. Az IE6 képes a cookie-k szűrésére is Hibajelentés: valószínűleg mindannyiunkkal megtörtént már, hogy böngészés kellős közepén az IE beadta a kulcsot. Az IE6-ban lehetőségünk van a hibát a Microsoft felé jelenteni. Ha a hiba ismert, a válasz akár egy javítócsomag vagy hotfix címe is lehet

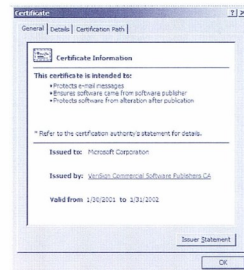
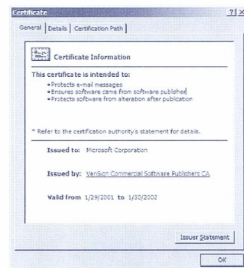


## Eső után köpönyeg :-)

### Baklövés a VeriSign-nál

A VeriSign Inc., a világ egyik legnagyobb digitális tanúsítványokat kiadó szervezete értesítette a Microsoftot, hogy 2001. január 30-án és 31-én két VeriSign Class 3 tanúsítványt adott ki egy magát Microsoft alkalmazottnak valló egyénnek. A kiadott tanúsítványok szoftverkomponensek és scriptek digitális aláírására használhatók, és úgy tűnik, mintha azokat a „Microsoft Corporation” adta volna ki. A hamis tanúsítványok segítségével aláírt rendszerkomponensek például egy-egy dokumentum megnyitásokor a bennük található scriptek segítségével lefuthatnak, miközben a gyanútlan felhasználó azt hiszi, a Microsoft-tól származó programot futtat.

A VeriSign természetesen visszavonta a tanúsítványokat (*a visszavont tanúsítványok listája – Certificate Revocation List – már tartalmazza*), csakhogy a VeriSign által kiadott tanúsítványok nem tartalmazták a visszavont tanúsítványokat tartalmazó lista letöltési helyét (*CLR Distribution Point, CDP*). A javítás a Windows 95 óta megjelent összes Windows operációs rendszerekben futó IE4.01 SP2, IE5.01 SP1, IE5.01 SP2, IE5.5 SP1 böngészőkhöz készült, és a [3] címről tölthető le. A javítást a Windows 2000 Service Pack 2, az Internet Explorer 6 és a Windows XP Gold (*gyártásra kész, végleges*) verziója tartalmazza majd, ezek megjelenéséig az érintett böngészők újratelepítése vagy rendszerfrissítés (*W2000-re, WXP-re*) után a hotfixet újra kell telepíteni. Ha a javítást még nem telepítettük, az ellenőrzést kézzel is elvégezhetjük. A digitálisan aláírt programok futtatása előtt megjelenő figyelmeztető dialógusablakban ellenőrizzük a tanúsítványokat. (*Kattintsunk az aláíró nevére*). Ha a tanúsítvány kiadási dátuma 1/29/2001 vagy 1/30/2001, a tanúsítványkiadó (*Issued by*): „VeriSign Commercial Software Publishers CA”, a tulajdonos pedig (*Issued To*): „Microsoft Corporation”, akkor a tanúsítvány hamis, ne futtassuk a programot, amirez „ragasztották”. Ezeket a napokon a Microsoft részére nem készült valódi tanúsítvány.



• **Íme a két hamis tanúsítvány. A dátumot figyeljük!**



Végül két megjegyzés:

- ☞ A Microsoft soha nem küld levélben programot (*csak tiszta forrásból származó szoftvert futtassunk*)
- ☞ A digitális tanúsítvány kiadása előtt a tanúsítványt kérő személy személyazonosságának ellenőrzése a tanúsítványkiadó szervezet dolga.

A témáról részletesen a [4] címen olvashatunk.

#### XP-hírek

Mindenekelőtt: itt a Windows XP béta 2, amihez a hivatalos tesztlők, és a fejlesztők máris hozzájuthatnak. Az MSDN Professional és Universal előfizetők letölthetik a webről, vagy várhatnak a következő CD-szállítmányra. A Windows XP béta 2-t már aktiválni kell, az MSDN előfizetők ehhez egy 10 aktiválás erejéig érvényes kódot kapnak.

A tesztlők immár a Windows XP béta 2-höz készült hardverkompatibilitási listát (HCL) is elérhetik [5]. Az érdeklődők látogassák meg a Windows XP hivatalos webhelyét a [6] címen, és ne hagyják ki a Windows SuperSite béta 2 tesztjét se! [7].



#### Jégeső!

A Microsoft március 19-én bejelentette a .NET vízió-család legújabb elemét, az intelligens, XML alapú webes megoldások fejlesztését támogató HailStorm (*kb. jégeső*) technológiát. A megoldás a már széleskörűen használt Microsoft Passport-ra alapul, és egyik fő célja az lenne, hogy egységesítse a felhasználók különféle webhelyeken történő azonosítását úgy, hogy a felhasználó adatai mindvégig a felhasználó tulajdonában maradnak, és ő dönthet azok felhasználhatóságáról. [8] A témáról későbbi számainkban részletesebben is szólnak majd.

# Microsoft .net

#### A cikkben található URL-ek:

- [1] <http://www.microsoft.com/windows/ie/preview>
- [2] <http://www.w3.org/P3P/>
- [3] <http://www.microsoft.com/downloads/release.asp?ReleaseID=28888>
- [4] <http://www.microsoft.com/technet/security/bulletin/MS01-017.asp>
- [5] <http://www.microsoft.com/hcl/default.asp>
- [6] <http://www.microsoft.com/windowsxp>
- [7] [http://www.winsupersite.com/reviews/windowsxp\\_beta2.asp](http://www.winsupersite.com/reviews/windowsxp_beta2.asp)
- [8] <http://www.microsoft.com/net/hailstorm.asp>

ADA STRA RT

<http://vilagokorseg.hu>

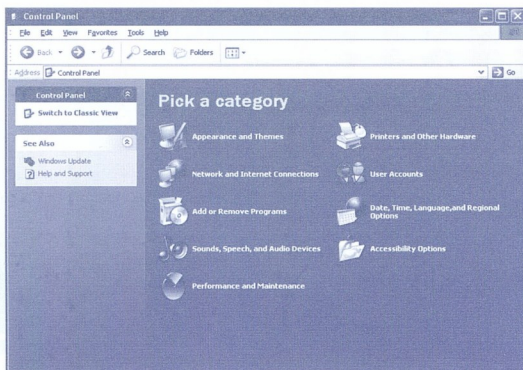




# Whistler

Sok érdekeset és jót hallottunk már a Windows XP, avagy a bennfentesebbek számára Whistler operációs rendszerről. Képernyőképek jelentek meg a Weben, melyek igazolják, hogy a számunkra, informatikusok számára oly érdektelen felhasználói felületen mi minden változott. Még szerencse, hogy a csicsmicsa kikapcsolható lesz :)

Én azonban azért ragadtam billentyűzetet, hogy egy kicsit a lényegi dolgokról is szó essék, s így jelen cikkben kizárólag díszítési elemként használok a felhasználói felületről készített fotókat... Nem hiába telt el 1-2 év a Windows 2000 kódjának lezárása óta, hisz megint olyan új felhasználói igények és ötletek merültek fel, melyeket egy új operációs rendszernek hoznia kell.



☞ *Így néz ki a Control Panel a Windows XP-ben*

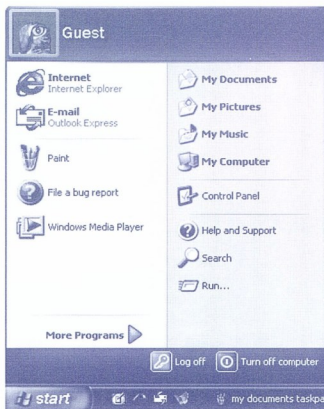
### Top-down

Haladjunk a felszínes izéktől a tényleg fontos szolgáltatások felé, tehát még mielőtt végleg elmerülnénk a rendszer mély rejtelmeiben, vegyük sorra, mi mindennel próbálnak minket elcsábítani a marketingesek.

☞ It van mindjárt a gyorsított rendszertöltés. Mivel Profissional-t nem is telepítettem, teljesen autentikus Advanced Server mérési eredményeim vannak a bootolás sebességéről: negyedannyi időbe telik, mint szegény „öreg” Win2K-nak. Ez már majdnem az OnNow sebessége, mely kezdeményezés célkitűzése a bekapcsolás utáni azonnali használatba vétel elérhetővé tétele – s nem ám hibernálással!

☞ Azután ki ne emlékezne borzongással a Windows 2000 kezdeti néhány hónapjára, amikor semmilyen módon nem lehetett vele CD-t írni, mert ő ugye nem, de az Easy CD Creator sem, meg a Nero sem. Persze, most már könnyű! De ha megemlítem, hogy a Whistler magától tudni fogja a CD írást, azt hiszem érdekes részletet árulok el. S miért is ne írná? Gondolkodott már valaki azon, hogy miért

point CD-t nem tud írni kedvenc op-rendszerünk? ☞ User switching. Különösen az otthoni felhasználók fogják szeretni ezt a szolgáltatást, ahol a kényyszerű ki-be-jelentkezésnek biztonsági okai nincsenek, sokkal inkább arra használjuk a profilokat, hogy apu munkához használatos desktopját ne árásszák el a gyerek játékaiknak shortcutjai, a háttér bitmap különbözőségéről nem is beszélve. Ilyen környezetben a kijelentkezés felesleges nyűg, amely a munkakörnyezet elvesztésével jár, mikor a gyerek csak egy emailt akar írni. A User Switching lehetővé teszi, hogy (használaton kívül) a Terminal Serviceshez, s az ötlet is onnan jött) egy-nél több munkamenet álljon bejelentkezve készenlétben, amelyek között név+jelszó megadással lehet majd váltogatni. Egyelőre kell hozzá helyes név és jelszó, de Redmondban, a Microsoft Research-nél már túlléptek ezen. Az Easy Living project keretében épült egy olyan ház, amelyk felismeri a lakóit, és tudja, hol vannak az épületben. Emiatt azt is tudja, hogy éppen ki, melyik családtag veszi éppen kezébe a háztartás egyetlen, drótnélküli billentyűzetét – és már switcheli is a kívánt felhasználói felületet. Még pár év és ez is a miénk lesz :)



☞ *A Start menü is megváltozik*

És most következen valami félkomoly...

### Active Directory 2.0

Noha az Active Directory mára már fényesen bebizonyította stabilitását és méretezhetőségét, azért azt hozzá kell tennünk, hogy kényelmes használatához hiányzik még egy-két dolog. Például nem lehet az objektumokat drag-and-drop módszerrel áthelyezni másóhova. Az még csak hagyján, hogy másik OU-ba nem így kell, mert azt legalább lehet. De másik

tartományba, fába, erdőbe semmilyen egerészós módszerrel nem lehet! Hacsak az utólag telepíthető Active Directory Migration Toolt nem tekintjük egerészós megoldásnak. De nem az. Szigorú tervezést igényel minden tétel, amire ezzel az eszközkel vetemedünk

– A Whistler mindezt tudja.

Azután ott van a rendszergazdák életének megszorítására az a fantasztikus lehetőség-hiány, hogy egyszerre egynél több user-en változtassak bármit. Nem megy! Amint egynél többet jelölünk ki, eltűnik a Properties menüpont! Sic! Jöhet a NetAcademia script tanfolyam, és máris ezresekkel kezelhetjük az objektumokat :)

– A Whistler ezt is tudja.

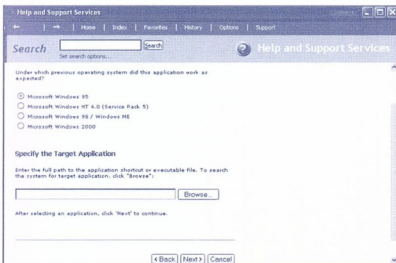
No meg a Global Catalog rémálom. Manapság ha nincs GC, nincs bejelentkezés. Még akkor is így van ez, ha a GC-nek nincs is értelme, mert egytartományos a környezet, ráadásul Universal csoportokat sem hoztuk létre.

– A Whistler-ben ezt is megoldották.

És ugye az ominózus csoportobjektum, melynek egyetlen, gígszavú dagadó attribútuma hordozza a csoport tagsági információit. Ennek egyes következménye, hogy egyidejű csoporttagság-módosításokkal könnyen adatvesztésre tud jutni két rendszergazda, hisz a replikáció nem tud finomabb egységet replikálni, mint egyetlen attribútumot. Nem mintha magyar ember valaha az életben találkozna ezzel az egyébként könnyedén elkerülhető replikációs problémával, de azért ami csúnya, az csúnya.

– Némi sémamódosítás árán ez a szerencsétlenség is begyógyul a Whistler-ben.

Sok jelenlegi parancsori eszköz grafikus arcot kap, úgyhogy tovább fog dagadni az Administrative Tools menüpont. Azt hallottam (*de látni nem láttam*), hogy az NTDSUTIL kap majd grafikus arcot, ami meg fogja könnyíteni az igazán advanced Active Directory feladatok végrehajtását. Egérrel lehet majd többek között offline töredettségmentesítést végezni, a Configuration partíciónál törölni, Authoritative Restore-t csinálni stb.



### ☛ A Whistler kompatibilitás-varázslója

Most pedig beszéljünk valami igazán komoly dologról...

### Komoly újdonságok az AD-ben

A madarak csírpelése szerint az Active Directory-n nemcsak ráncfelvarrást hajtanak végre, hanem bele is nyúlnak jócskán.

☛ Például lehetővé fog válni átmeneti, illékony objektumok és attribútumok létrehozása, melyeket eddig nem illett az AD-ben tárolni, hisz minden adat huss, kiment

az NTDS.DIT fájlba, majd szétreplikálódott. Az átmeneti objektumok replikációja ki/be kapcsolható, és csak memóriában vannak, azaz nem fogják össze-vissza tördelni az NTDS.DIT-et állandó létrejövőnkkel és törlődésükkel. Ezzel megnyílik az út olyan adatok központi kezelésére, amelyekről eddig csak áldomtunk:

☛ Végre látszódhat az AD-ben, hogy egy felhasználó vagy gép be van-e jelentkezve, mert fel lehet végre venni az attribútumok közé, hogy a gépnek van-e Secure Channelje (*azaz be van-e kapcsolva*), a usernek meg van-e Kerberos jegye

☛ Végre nem lesz fél-legális az Active Directory Integrated DNS zóna

☛ Stb.

☛ A meglévő három egész (*plusz egy megtűrt*) partíció (*Schema, DC, Configuration, +1 GC*) mellé az alkalmazás-fejlesztők maguk is készíthetnek partíciókat, melyeknek replikációs viselkedését ők maguk írhatják majd le!

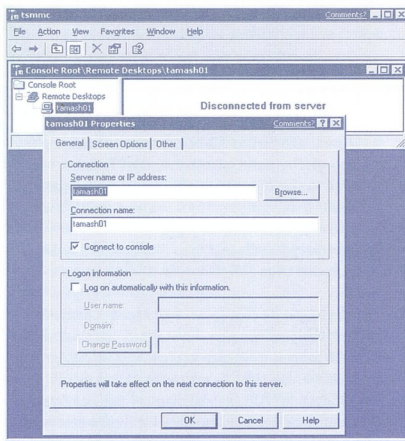
☛ Virtual list view (VLV). Valami olyasmi, hogy az egy, megcsontosodott OU hierarchián kívül más nézőpontból is rácsodálkozhatsz majd az AD-re. Azaz nézetet válthatsz. Unom az OU hierarchiát, hadd lássam más szerint felülviz!

### A fej nélküli lovas

Érdekes megoldásnak tűnik számomra a headless operation, magyarul orcátlan futás. Ez azt jelenti, hogy az elindított Whistler-nek egyáltalán nincs képernyője, billentyűje, egere. Mi több, a headless futáshoz ki kell szedni a videokártyát a gépből, mert ha a rendszerindítés talál ilyen, használata veszi. Azaz a headless működéshez megfelelő BIOS is kell, amely elviseli a videokártya hiányát. Mire jó mindez? Hogy a kiszolgálót betehessük az ellenség hűtőkamrájába, a Un\*xok közé. Persze ehhez más is kell: teljes körű rendszerfelügyeleti lehetőség hálózaton keresztül. Nem, nem a Terminal Services, mert azzal egy-két dolgot nem lehet megcsinálni, s azért sem, mert ha a hálózatot konfiguráljuk halálba, akkor a Terminalozásnak lóttak. Nem úgy a telnetelésnek, ami akár TCP/IP-n, akár soros porton rendelkezésre áll. S szűgyenszeme, annyit év után a Windows XP eljut oda, ahol a Un\*xok eddig is elváltak: teljeskörű felügyelet Telneten keresztül. Az öszvér neve: Special Administration Console. Működni még nem láttam.

### Performance Monitor

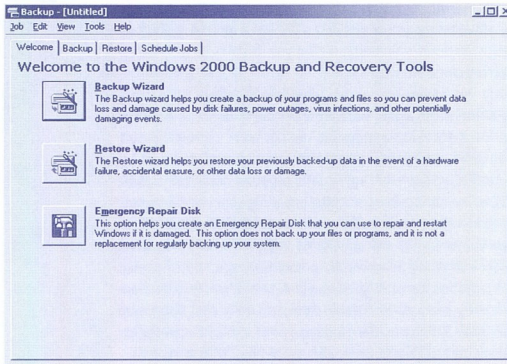
Hány éve várom már annak lehetőségét, hogy DELTA, azaz különbségszámlálókat nézegethessenek a Performance Monitorral! Évek óta mondogatom Bill-nek, és most végre hallgattott a jó szóra. Eddig ugyanis például semmilyen módon nem lehetett megállapítani egy adott program által felhasznált összes memóriát. Érdekes, hogy ezt eddig senki más nem vette észre rajtam kívül, talán senki nem használ PerfMon't? Szép vagyunk! Egy szó mint száz, egy processzről meg lehet tudni a Working Set méretét, a Pool Nonpaged Bytes-t, meg sok egyébét, de ezek közül egyik sem a processz teljes memóriafogyasztása! Ne vitatkozzunk, a Committed Bytes sem az! Össze kellene adni a Working Set-et a Pool Paged Bytes-sel, de hát ugye erre majd csak a Windows XP lesz képes.



☞ *Már a Professionalban is lesz Terminal Services!*

**Automated System Recovery (ASR)**

Visszatér! Ez a feature egyszer már megjelent a Windows 2000 Beta 3 változatában, de a végleges kódból kivették, mert nem készült el hibátlanul a Win2K piaca dobásáig. Miről is van szó? Operációs rendszer mentést az élő operációs rendszer segítségével, NTBACKUP.EXE-vel szoktunk végezni, s ez nem is zavar senkit. Annál zavaróbb, hogy az így készített mentést kizárólag élő Windows 2000-rel lehet visszatölteni, hisz ki más ismerné fel az NTBACKUP által írt szalagot, mint az NTBACKUP?! Azaz kell egy oprendszer az oprendszer visszaállításához. Agrém. Ezt régóta tudjuk, s a fejlesztők réges-régen hozzákézdtek egy olyan megoldás kidolgozásához, amikor a mentés nem csak egyszerűen szalagot ír, hanem mellé készít egy bootolható csoda-flopi-t, amiről ha indítottunk egy szűz gépet, bekéri a szalagot, és feltornászza a gépre az oprendszert. Ez az Automated System Recovery, és most bebizonyítom, hogy a Windows 2000-ben már majdnem benne volt. Az alábbi ábra az NTBACKUP-ról készült. Figyeljük meg az ablakot tüzetesen!



☞ *Hol az ASR hűlt helye?*

A három nagy gomb alatt simán elférne egy negyedik is! Az ott biza az ASR hűlt helye. A Beta 3-ban még ott volt. Ipartörténeti érdekesség? S hogy honnan tudom? Mert 1998-ban Win2K Beta3 tanfolyamokat tartottam az érdeklődő megoldásszállítóknak, és sűrűn dicsértem ezt a feature-t – amíg egyszer csak eltűnt.

Perse be kell vallanom, hogy ezt nem próbáltam ki a Whistlerben, úgyhogy még akármí is lehet, például más funkciója lesz, vagy megint kilopják az utolsó pillanatban. Az ASR-re egyelőre nem alapoznám a vállalat jövőbeni mentési stratégiáját...



☞ *Segítsé!*

Sok minden jöhet-mehet még Whistler ügyben, a jelenlegi Beta még csak nem is teljes funkcionalitású, sőt, a funkciókészlet terve sem végleges. Éppen döntés előtt áll az a kérdés, hogy vajon támogassák-e a NetBEUI és a DLC protokollokat a jövőben. Ha új tények merülnek fel a színen, ígérem,

Folytatjuk...

Fóti Marcell, MCSE, MCT, MCDBA  
Marcellf@netacademia.net



# Microsoft Network Monitor (V. rész)

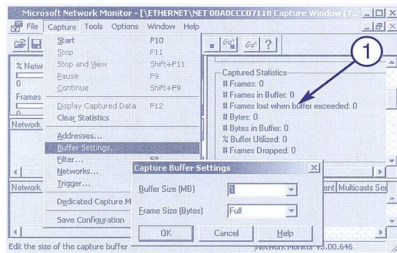


Az eddigiekben megismerkedtünk a hálózati felügyelés alapjaival, s a TCP csatorna felismerése megadta a kulcsot az összes rejtély megfejtéséhez, hisz minden izgalmas forgalom TCP csatornában zajlik. Ha megvan a csatorna építése és bontása, s ezzel maga a csatorna, akkor már „csak” a köztes hálózati forgalmat kell elemeznünk. Ehhez azonban új eszközökkel, és a mindenkinben szunnyadó matematikai képességek (sőt *Boole-algebra!*) kiaknázására lesz szükségünk. A dolog nem olyan véres, mint amennyire ijesztő: egyszerű logikai kapcsolatokkal fogunk felépíteni a csomagok egyes tulajdonságai alapján, azaz szűrünk.

De lassan itt az ideje, hogy elszakadjunk az eddigi vizsgálati módszerektől is, hisz eddig steril körülmények közt NetMonoztunk: csak az a hálózati forgalom került a szemünk elé, amit valóban szemügyre akartunk venni. A valós életben, amikor a Network Monitort hibakeresésre, és egyéb véres dolgokra használjuk, sajnos nincs lehetőség az összes nemkívánatos hálózati forgalom elnémtására. Mi több, nemcsak kiszámítható, és adott esetben elnémtatható hálózati forgalmakkal szembesülünk, hanem olyasmivel is, ami – csak úgy – egyszer van, egyszer nincs. Ki tudná ugyanis pontosan megjósolni, hogy két tartományvezerlő mikor kezd nász tánca, mikor indul el a replikáció közöttük.

Megoldás szerencsére majdnem mindig van. Ha a felvételi körülmények nem ideálisak, az ügyes nyomozó mindig mindent begyűjt, s majd (othon, a *kandalló melegénél*) offline szétválogatja az értékes nyomokat az értéktelenektől. Ezt az eljárást szűrésnek nevezzük. Az igazat megvallva a NetMon képes lenne eleve szűrt adatok felvételére, mert futás közben is használhatunk szűrőket, de ezzel egyfelől lehet, hogy értékes adatoktól fosztjuk meg magunkat, másfelől a futás közbeni szűrők sokkal primitívebbek, mint amit offline használhatunk.

Érdemesebb talán egy kicsit több memóriát rászánni a feladatra, s egy kicsivel többet összegyűjteni, több szeméttel több az arany – mondja az ózsanszkrit közmondás. A Network Monitor memóriában tárolja az elkapott hálózati forgalmat, ehhez indúlaskor puffert foglal (*alapértelmezésben 1 megabájtot*), s a pufferen jár körbe-körbe. 1 megabájt (*terhelt hálózat esetén!*) általában kevés, hisz ezen a területen körbe-körbe jár, így elég nagy eséllyel felülírja a 30 másodperccel korábban rögzített adatokat. Az alábbi ábra azt mutatja, hogyan lehet a Network Monitor alapértelmezésben 1 megabájtos memóriapuffert megnövelni – mondjuk – tíz megabájtra.

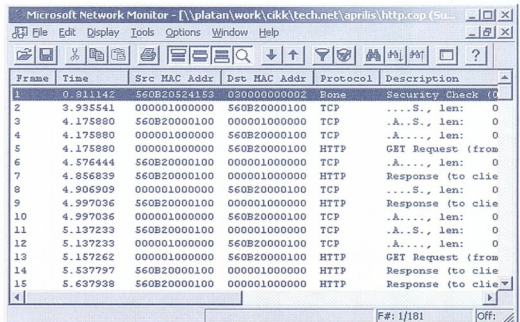


☞ **Növeljük meg a puffert, hisz 1 MB semmire sem elég!**

A fenti ábrán az is megfigyelhető ①, hogy a puffertülszordulás nem maradna rejtve árgus tekintetünk elől – ha árgus lenne a tekintetünk. Én már elemeztem körbefordult hálózati forgalmat. Másfél órát rabolt el az életemből ez az apró figyelmetlenség.

## Webbögészés

Első hálózati forgalmunk a mai alkalommal a http.cap, melynek tartalma a <http://www.netacademia.net> letöltése közben keletkezett. Ez sem nevezhető különösebben zavarosnak, de több TCP csatorna van benne, így minimális szűrőzésre alkalmas ad. Ha letöltik kedves olvasóm is [1] címről, és beletekintene, valami hasonlót kell látniuk ehhez:



☞ **A http.cap tartalma**

Vagyis a szokásos Bone protokoll mellett (*mely a Network Monitor használat velejárója*) csatornaépítés és HTTP parancsokat láthatunk, összesen 181 csomagot keresztül. Ugye mostanra már mindenki tudja, hogy amit HTTP-ként látunk a NetMonban, az csak az adott csomag lényege, de valójában mindegyik sorunk összetett csomagokat tartalmaz, a HTTP – mint tudjuk – TCP-ben utazik, a TCP-t IP szállítja, az IP pedig egy Ethernet keret hasznos adata.

### Szűrési lehetőségek

Vizsgáljuk meg a NetMon szűrési képességeit! Talán nem lesz meglepő, hogy különböző szintek információi alapján képes szűrni: kiválogathatjuk egy adott MAC Address hálózati forgalmát (az ingyenes NetMon használok eleve csak a saját forgalmakat (és a Broadcastokat) látják), egy adott IP című gép adatait, valamint protokollok szerint is szűrhetünk.

#### Fontos!

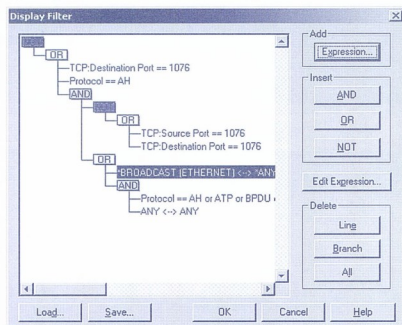
**A MAC Address-szintű szűrés félreértésekre adhat okot, hiszen távoli hálózaton lévő gépek esetén nem a távoli gép MAC Addressét látom, hanem a Default Gateway hálókártyájának innenső lábát!**

#### Miért?

**Mert a MAC Address nem alkalmas végponttól – végpontig történő címzésre, erre a feladatra az IP való. A kemény félre diagnosztizálások elkerülése érdekében szűrjünk inkább a feladó és címzett IP címére!**

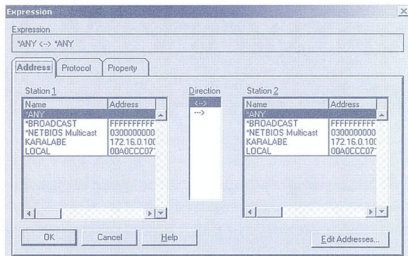
A NetMon becsapós kettősségének jegyében a szűrés (filter) elérhető a főképernyőn is, és az elkapott forgalmak ablakában is. A két filter máshol van a menüben, másképp néz ki, és máris is való! A főképernyős filter (Capture menü->Filter) egy lényegesen kisebb funkciókészletű ablakot hoz elő, melyben a futás közbeni szűrés beállításait csodálhatjuk meg (de most nem használjuk), míg a forgalmi adatok ablaknál található filter (Display menü->Filter) teljeskörű, ámde offline szűrés végző, magyarárn elrejti/megjeleníti az általunk kiválasztott csomagokat. (Emlékszik még valaki a szűrésre? Használjuk azt is bátran!) Mielőtt teljesen belemerülénk, had hívjam fel a figyelmet az eszközsáv gombjára, ezzel ugyanis mindig visszatérhetünk a szűrésmentes nézethez, nem feltétlenül kell újrainstallálni ehhez az egész Windowst :-)

A gazdag funkciókészletű filterablak így fest:



#### • A teljeskörű filterablak egy bonyolult szűrőkifejezéssel

Aki ingyenes NetMont használ, az az ablak megnyitásokor figyelemztetésben részesül, hogy nála a szűrés már kissé megtörtént, csak a saját gépének forgalmát látja. Ebben az ablakban – mint azt talán a gombok is sejtetik – hihetetlenül bonyolult szűrési feltételek adhatók meg, melyeket azután el is menthetünk. Most nyomjuk meg az Expression... gombot, hogy meglássuk, milyen építőközből válogathatunk a bonyolult logikai kifejezések összerakolásához:

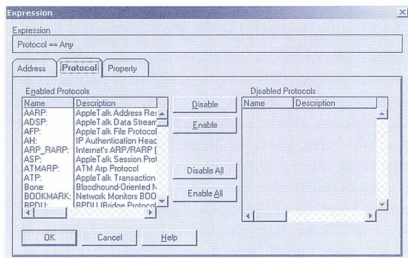


#### • Szűrhetünk címekre, protokollokra és egyes bitekre is

Jól látható, hogy háromféle szűrési lehetőségünk van.

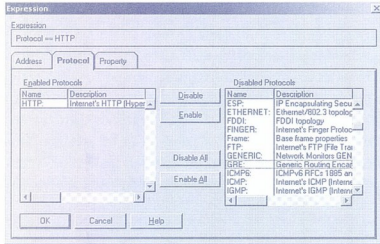
- Az első (Address) fül segítségével MAC Addressre, IP és IPX címre, valamint egyéb, a NetMon által felfedezett címekre szűrhetünk. Alapértelmezésben az összes elkapott hálózati forgalmat láthatjuk. (ANY<->ANY)
- A második (Protocol) fülön választhatjuk ki a megjeleníteni kívánt protokollok listáját. Alapértelmezésben az összes előforduló protokoll megjelenített.
- A harmadik (Property) fülön pedig az egyes protokollok belső információi alapján is szűrhetünk, ezt néhány bekezdéssel később a TCP csomaga megragadására fogjuk használni.

A cím szerinti szűrést külön példán nem mutatom be, annyira egyértelmű. Lássuk a protokoll szerinti szűrést! Ezen az első feladatot a http forgalom kiszűrése, ennek érdekében kattintsunk a második fülön, ahol a bal panelen látjuk azon protokollok felsorolását, melyek jelenleg átmennek a szűrőn – azaz mindent.



#### • Alapértelmezésben az összes protokoll engedélyezett

Ne sajnáljuk a Disable All nyomógombot, és dobjuk át a díszes társaságot a jobb oldalra, majd kegyelmezzünk meg a HTTP-nek, és azt az egyet kijelölve, nyomjuk meg az Enable gombot. Ezzel egy olyan szűrőhöz jutunk, mely csak és kizárólag a HTTP csomagokat jeleníti meg. Beállításunk eredménye így nézzen ki:



☞ **Így már csak a HTTP marad fenn a rostán**

Az OK gomb megnyomása után pedig a helyes filter valami ilyesfémn alakul *(ha nem ezt látják, a szűrés nem ugyanaz az eredményt fogja adni, mint nálam, és a későbbiekben nem egyről fogunk írni/olvasni):*

```
ANY
  -> Protocol == HTTP
  -ANY <-> ANY
```

Innen csak ez kizárólag HTTP Get és Response marad előtűnik, mégpedig egy-egy Get-hez hat-nyolc Response. A mértet mindenki tudja: Míg a Get elfér egy csomagban *(maximum 1514 bájtt)*, a válasz már ritkán. Nézzük meg a legelső Get-et *(hatodik csomag a 181-ből)*:

**HTTP: GET Request (from client using port 1075)**

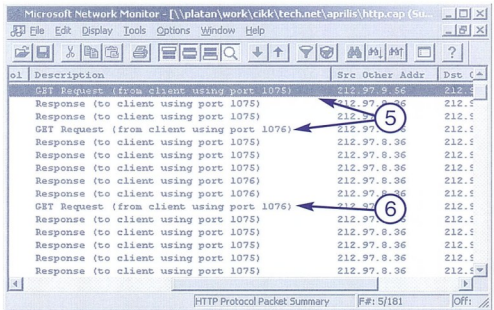
```
HTTP: Request Method = GET
HTTP: Uniform Resource Identifier = /
HTTP: Protocol Version = HTTP/1.1
HTTP: Accept = */*
HTTP: Accept-Language = en-us
HTTP: Accept-Encoding = gzip, deflate
HTTP: User-Agent = Mozilla/4.0 (compatible;
MSIE 5.5; Windows NT 5.0; COM+ 1.0.2204)
HTTP: Host = www.netacademia.net
HTTP: Connection = Keep-Alive
HTTP: Cookie = QQ=hu;
```

☞ **ASPSESSIONIDGQGQGBDY=OHFOADBOBJAHCFNBADJKAKE**

☞ A ② jelzésnél olvasható a kért fájl neve, a sima perjel (/) utal a default dokumentumra. A default dokumentum a mi esetünkben default.asp névre hallgat, és a következő, hetedik csomagban kapjuk vissza. Valószínűleg mindenki számára ismerős a weblapok letöltődésének menete: elsőként egy HTML *(vagy ASP vagy egyéb)* kiterjesztésű szöveges fájlt tölt le a böngészőnk, melyből kiolvassa, hogy milyen egyéb dokumentumokra lesz még szükség a megjelenítéshez: képekre, stíluslapra stb. Ezeket a további komponenseket külön HTTP Get parancssal szedegeti le szépen, egyesével, amire bizonyíték a képek későbbi megjelenése, illetve a www.netacademia.net esetében egy komoly pillanatig stíluslap nélkül látszik az oldal, majd egy másodpercre rá megérkezik a stíluslap, és a böngésző át-szabja a dizájnt *(ez a jelenség csak a legelső látogatóknak szűrhat szemet, mert a stíluslapfájl éppúgy bekeskellődik, mint a többi hasonló adat)*. Ha a többi Get-et megvizsgáljuk, látszik is a letöltés: a 13. csomagban a /css/main.css stíluslap, a 22.-ben pedig az /images/hatter2.gif stb.

☞ A ③ jelnél láthatjuk a protokoll verziószámát. A HTTP 1.1-nél tart. Talán a legszembetűnőbb különbség az 1.0 és az 1.1 között a TCP csatornák újrafelhasználásában mutatkozik. Míg az 1.0 minden egyes dokumentum letöltéséhez külön csatornát nyit, s ezzel csomagotmegeket indít újtjára, addig az 1.1 csak annyi csatornát nyit, amennyi a letöltés párhuzamosasá tételének érdekében szükséges – vagy más szóval annyit, amennyit a sáv-szélesség elbírn. Minek is nyitna objektumonként külön csatornát, ha a sáv-szélesség amúgy sem teszi lehetővé egyidőben egynél több GIF letöltését?

☞ Végül a ④ jelle hívárn fel a figyelmet. Itt utazik ugyanis az ASP Suli kukija! Az ASPSESSIONIDGQGQGBDY nevű változó csak legelőre komplett összevisszaság, a túrlemez szelődő felfedezheti benne az ASP Session ID szavakat.



☞ **Az 56K már elegendő sáv-szélességet biztosít két párhuzamos csatorna megtöltésére**

A fenti ábra ⑤ jelénél látható, ahogy a böngésző – ki sem várva a 1075-ös csatorna eredményét – párhuzamosan újtjára indít egy másik kérését a 1076-os csatornában. A ⑥ jelnél pedig a csatorna újrahasonosítására látunk példát: a HTTP 1.1 nem lebontja a dolga végzett csatornát, hanem új feladatot ad neki! Sáv-szélesség-spórolás: 6 csomag *(3 csomag a bontásra + 3 csomag új csatorna építésére)*! Milyen különös, hogy a HTTP 1.1 pont akkor szabadít fel jelentős sáv-szélességet, mire az egész sáv-szélességkérdés röhejessé válik – valahol Amerikában...

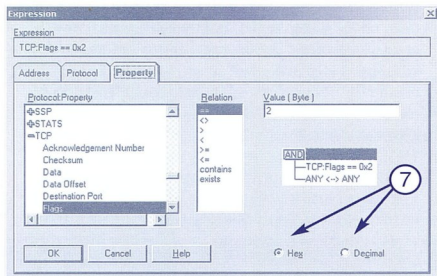
**Hány csatorna van?**

A fenti képernyőképen észrevettünk két csatornát. Ennél azonban tudományosan is megállapíthatjuk, vajon e fájlban hány különböző TCP csatorna rejtőzött! Egy olyan szűrőre volna szükségünk, amely csak a TCP csatornák letelejét, építési szakaszát mutatja meg nekünk, s azon különböző meg-számálhatjuk, vajon tényleg kettő van-e. Ehhez a szűrőhöz bizony már a TCP bitjeire kell szűrünk, mert a three-way-handshake-t *(chip-chip-choka)* kell elcsipünk:

```
...S.
.A...S.
A....
```

Az S flag egyetlen bit csupán a TCP fejlék mezőjében! Ennek kiszűrésére szolgál az Expressions ablak harmadik füle, ahol le lehet menni – nos, sajnos bitszintre nem. De bájtt-

sztíngt mindenképpen. Ahhoz azonban, hogy sikerrel járjunk, meg kell állapítanunk a S bjtértéket. Ehhez manuálisan kikeresünk egy S csomagot, és onnan kiolvassuk, hogy S=2. A szűrési feltétel ezek után így adható meg:

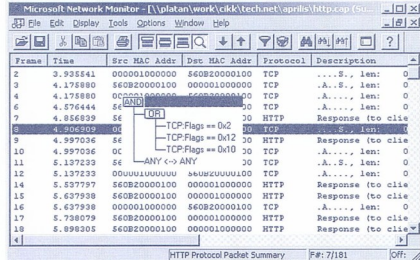


» Az S, azaz SYNchronize Sequence Number-ek kiszűrése

A fenti ábra üres helyére odacsempészttem a szűrő logikai felelőjét, hogy könnyebb legyen ellenőrizni a helyes beállítás. Az eredmény nem meglepő, két TCP csomagunk van. A szűrő két csomagot hagyott meg (ezért e képernyőképet nem is mutatom), így sajnos nem látszik a teljes csatornaépítés. Azonban ha kiléssük a többi lépés bjt-kódját is

- ...S. = 0x2
- A..S. = 0x12
- A.... = 0x10

Akkor megfelelő büvészkedés után majdnem meg is kapjuk munkánk gyümölcsét (az ábra közepére varrtam a logikai kifejezést, már OR is van benne!):



» Sok büvészkedés után kevés eredmény: a TCP „becsapott”!

Az ábra tanúsága szerint kétségkívül megfogtuk a three-way-handshake-ke-t – de egy kicsit túl nagyot markolt ez a szűrő. Hogy miért? Mert

A....

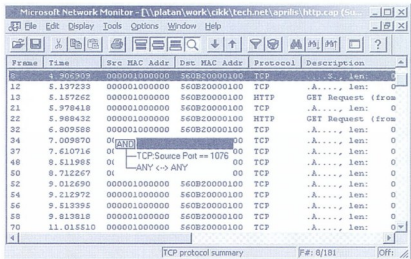
mindenütt van! Miközben mi a HTTP-re összpontosítottunk, talán el is felejtettük, hogy hűségees abszolgánk, a TCP a hát-térben csatornakarbantartási munkát végez, ACK-ol vesztettül mindkét irányba!

**Fogjunk meg egy csatornát!**

Végeztül legyen a feladat egyetlen TCP csatorna kiszűrése, hisz ez a feladat eléggé mindennapos ahhoz, hogy külön megvizsgáljuk.

Ha elejétől a végéig meg szeretnénk ragadni a http.cap fájl-ból mondjuk a fent említett 1076-os csatornát, akkor port-számzűrést kell használnunk. A 1076 ugyanis nem más, mint az Internet Explorer által használt kliensport a webkiszolgáló elérésére. Mi sem egyszerűbb! Készítsünk filtert a TCP protokoll Source Port mezőjére, ahol a portazonosító 1076!

Akinek semmi sem maradt a szűrés után, az elvettette a hexa – decimális átváltást (7). Akinek azonban így fest az output...



» Szűrés a 1076-os portra. Felsiker.

...az pontosan ugyanazt a hibát követte el, mint én. A MAC Address-ből úgy tűnik, sikerült egyirányúvá szűrniünk a forgalmat, amit bizonyít, hogy egy fia HTTP Response nem maradt a képernyőn. Mi lehet a baj?

A tűzfalnakl már szóba került, hogy hányféle forgalom szület-het két portazonosító felhasználásával. A hiba oka az, hogy a HTTP Response csomagokban nem a feladó portszáma a 1076-os (a feladó a 80-as portról a webkiszolgáló), hanem a címzetté. Gyorsan javítsuk ki a szűrési feltételt, és máris miénk a 1076-os csatorna teljes forgalma! A helyes szűrőfeltételt így fest:



**Elkeserítésképpén**

Végeztül hadd lombozzak le mindenkit: előfordult már a praxisomban olyan hálózati forgalom, amit nem lehetett szűrés-sel különválasztani: egy Windows 2000 tartományvezérlőn nem működött az Active Directory keresés LDAP névelődása (minden más gépen ment), ami igencsak nehezen lehetett volna elválasztani a többi LDAP forgalomtól. Meg is gyűlt ve-le a bajom! Nincs mit szűrni, ha a DC saját hálózati forgalmában, és az Active Directory keresésben is azonos a feladó és a célgép IP címe, ráadásul azonos protokollt használnak! Nem tehettem mást, lestem, hogy mikor hallgat már el a DC, hogy végre az én forgalmam kivethető legyen a csatazajból.

Fóti Marcell, MCSE, MCT, MCDBA  
marcellf@netacademia.net

**A cikkben szereplő URL-ek:**

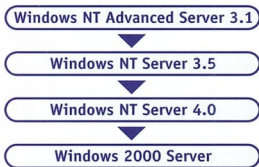
[1] <http://technet.netacademia.net/feladatok/netmon>

# A kernel-architektúra fejlődése



A Microsoft Windows 2000 számos különböző rétegből épül fel, melyek együttesen alkotják az operációs rendszert. A rendszer magja a kernel. A kernel nem csupán egy a folyamatok közt, hanem speciális jogosultságokkal rendelkezik a hardver fölött. A kernel felelős a memória kiosztásáért az egyes folyamatok közt, az eszkökezelőkkel való kommunikációért, valamint a folyamatok ütemezéséért.

Az alkalmazásoknak a kernel biztosítja a működésükhöz szükséges memóriát, processzoridőt, és egyéb hardvererőforrásokat. A Windows 2000 Server kernel a Microsoft Windows NT 4.0 kernel továbbfejlesztésével jött létre, emez pedig a korábbi Windows NT kernelekből származik, ahogy az az alábbi ábrán látható. Az újabb verziók mindig a felhasználók, rendszergazdák és fejlesztők igényeinek figyelembevételével készültek.



## » A Microsoft Windows 2000 Server számos generáció fejlődésének eredménye.

Amikor a fejlesztőknek olyan funkciókra lett volna szükségük, amit a kernel nem tartalmazott, akkor a Microsoft az ő igényeiknek megfelelően módosította a kódot. Ilyen eset volt például az Internet Information Services 5.0 (IIS) rendszerhez. A kernel most már fejlett többfelhasználás támogatását is tartalmaz, mivel lehetővé teszi a Terminálszolgáltatás telepítését – nagyobb rendszer módosítás nélkül.

A Microsoft számos módosítást végzett a kernelen a rendszergazdák igényeinek megfelelően is, többek közt a Windows 2000 Server sokkal jobban skálázható mint a Windows NT korábbi verziói. A hálózati alkalmazások megbízhatóbban, hatékonyabban működnek, mint bármikor korábban. A különböző nemzetek felhasználói könnyebben használhatják az alkalmazásokat saját, megszokott nyelvi környezetükben.

Fontos megérteni a kernelen végzett módosításokat, mielőtt a különböző szolgáltatások fejlődését vizsgálánk. Nagyon sok szolgáltatás fejlesztése szorosan összefügg a kernel módosításával.

### Processzorkódoták és számlálás

Általános gyakorlat az Internetszolgáltatók körében, hogy egyetlen webkiszolgálót sok ügyfél közt osztanak meg. Eddig előfordulhatott, hogy egyetlen ügyfél weboldala a processzor idejének nagy részét lefoglalhatta, csökkenve a többi ügyfél oldalának teljesítményét is. Hasonló módon, egyetlen felhasználó rosszul megírt ASP kódja a teljes intranetkiszolgálót

használatatlanná tehetette. Hiába volt képes az IIS a sávszélesség szabályozására, ha maga az operációs rendszer képtelen volt az egyes weboldalakra jutó processzoridőt korlátozni. Két új lehetőség segít a többfelhasználós rendszerek hatékonyabb kihasználásában: a processzoridő számlálása és annak szabályozása. A processzoridőszámlálást az IIS alkalmazza annak rögzítésére, hogy az egyes webkérések mennyi processzoridőt vesznek igénybe. Ez lehetővé teszi a processzor használat alapján történő számlálást, továbbá a fejlesztőket is segíti annak kiderítésében, hogy melyek azok az oldalak, ahol az optimalizálás különösen hasznos lehet.

A technológia, ami a fenti két funkciót lehetővé teszi, a munkaobjektum. Hogy megérthessük miként működik egy munkaobjektum, vizsgáljuk meg, hogy az alkalmazások és szolgáltatások miként használják a folyamatokat. A szolgáltatásokat (mint például az IIS) sok folyamatot indítanak el annak érdekében, hogy a különböző feladatokat egyszerre tudják végrehajtani. Míg a sok folyamat alkalmazása növeli a teljesítményt, ugyanakkor nehezebbé teszi annak megállapítását, hogy melyik folyamat melyik feladathoz tartozik. Mivel az IIS 4 támogatja azt, hogy a virtuális kiszolgálók közös memóriaterületet használjanak, szinte lehetetlen megállapítani vagy szabályozni, hogy az egyes virtuális kiszolgálók mennyi erőforrást használhassanak.

A munkaobjektum lehetővé teszi azt, hogy az operációs rendszer az egy csoportba tartozó folyamatokat egyetlen egységként kezelje. Ezáltal sokkal egyszerűbb az alkalmazásokhoz tartozó erőforrást használatának figyelembe és szabályozása. Ez különösen fontos olyan esetekben, amikor több ügyfél használja ugyanazt a kiszolgálót.

### Spinzámlálás

A spinszámlálás egy olyan eljárás, ami többprocesszoros rendszerek teljesítményét növeli olyan esetekben, amikor több program kívánja ugyanazt az erőforrást egyszerre használni. A spinszámlálás határozza meg, hogy egy folyamat hányszor próbáljon egy erőforrást használni, mielőtt várni kezdene. Például nézzünk egy alkalmazást, ami egyszerre több lekérdezést futtat több processzoron. Az első lekérdezés megpróbál írni egy adott tábla adott sorába, de nem tud, mivel a második lekérdezés zárta azt a sort. Spinzámlálás nélkül az első lekérdezés vár egy ideig, blokkolva a lekérdezés végrehajtását. Spinzámlálás alkalmazásával az első lekérdezés egyszerűen újra megpróbál írni az adott sorba néhány alkalommal annak reményében, hogy a második folyamat gyorsan megszünteti a zárolást, és csak akkor fog írni, ha ezek a kísérletek sikertelenül végződnek.

Ez csak többprocesszoros rendszerekben hasznos, ott ugyanis egyszerre több folyamat futhat. Egyprocesszoros rendszeren a különböző folyamatoknak mindenképpen várniuk kell egymásra, így az ismételt hozzáférési kísérlet nem lehet sikeres, amíg az azt blokkoló másik folyamat meg nem kapja a vezérlést. Az egyprocesszoros rendszeren futó spinszámlálás alkalmazások nem lesznek lassabbak – csupán nem futnak majd gyorsabban.

### Scatter/Gather I/O

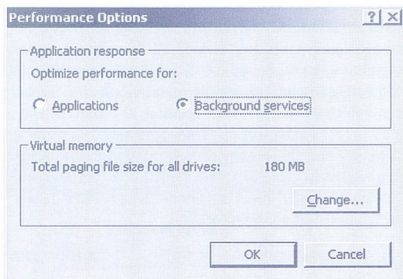
A Scatter/Gather I/O rendszer a Windows 2000 Server olyan újdonsága, ami a hálózaton lévő alkalmazáskiszolgálók teljesítményét javítja. Ennek használata az rendszergazda számára sem látható, mivel az alkalmazások minden további konfigurálás nélkül képesek használni. A technológia lehetővé teszi, hogy a memóriában elszórtan elhelyezkedő adatokat a lemezre folyamatosan lehessen kiírni. Az alkalmazásokat fel kell készíteni ennek használatára, ezért a jelenlegi alkalmazások működését ez a módszer nem javítja.

Ez a technológia először a Windows NT rendszerhez kiadott service pack csomagban jelent meg, speciálisan a Microsoft SQL Server teljesítményének javítására. A Windows 2000 Server az első olyan Microsoft operációs rendszer, ami ezt a technológiát alapfunkcióként tartalmazza.

### Időzítők

Az időzelet a szálak egy jellemzője, ami azt határozza meg, hogy az adott szál mennyi ideig futhat, mielőtt a vezérlés a következő szálra kerülne. Mivel a környezetváltás bizonyos időt vesz igénybe, egyes alkalmazásoknál előnyös lehet a kvantumidőt megnövelni, habár ettől a rendszer többfelhasználós teljesítménye csökkenhet.

A kvantum típusa lehet fix vagy változó hosszúságú. Ezt a Performance Options párbeszédablakban lehet beállítani, ami a System Properties ablak Advanced fülénél található.



#### ► Megadhatjuk a kvantumok típusát és hosszát.

Az alkalmazásoknak adott magasabb prioritás rövid, változó hosszúságú kvantumot, és sima, gördülékeny többfelhasználós működést eredményez. A háttérben futó szolgáltatások prioritását emelve hosszú, fix kvantumokat kapunk, ami a hálózati szolgáltatások teljesítményét javítja. Alapértelmezés szerint a Windows 2000 Server a háttérben futó szolgáltatásoknak, a Windows 2000 Professional pedig az alkalmazásoknak ad nagyobb prioritást.

### Windows Driver Model

A meghajtó egy olyan program, melynek segítségével az operációs rendszer kommunikál egy hardvereszközzel. A különböző videokártyák különböző képességekkel bírnak, és különböző módok kommunikálnak; a meghajtó végzi a „fordítást” a Windows 2000 és a videokártya között. Minden hardverhez szükség van meghajtóra: hálózati kártyákhoz, SCSI vezérlőkhöz, modemekhez éppen úgy mint lapelvasókhoz vagy nyomtatókhoz. A Windows korábbi verziói különböző meghajtókat

igényeltek minden operációs rendszerhez. Ez mind a meghajtókat író hardvergyártóknak, mind a rendszereket karbantartó rendszergazdáknek komoly munkát jelentett.

Az új Windows Driver Model (WDM) lehetővé teszi hogy a Windows 2000 és a Windows 98 rendszerek ugyanazt a meghajtót használják. Ez előnyös a hardvergyártóknak, mert nem kell két különböző meghajtót fejleszteniük. Előnyös a felhasználóknak, mert növeli a Microsoft rendszerek kompatibilitását.

A WDM-nek további előnyei is vannak. A Microsoft a meghajtóprogramok írásához szükséges munka nagy részét elvégezte. A hardvergyártóknak továbbra is kell írniuk egy kis minimeghajtót a termékeikhez, de ez lényegesen kevesebb programozói munkát igényel.

A WDM Kernel Streaming architektúra a valósidejű streaming anyagok teljesítményét növeli. A Windows előző verzióiban a streaming alkalmazásokhoz szükséges számítások nagy részét az alkalmazások felhasználói módban (*user mode*) végezték. Ezen műveletek nagy része most kernel módban fut, ami sokkal gyorsabb működést eredményez. Az alkalmazásokat fel kell készíteni a WDM Kernel Streaming használatára, ezért a régebbi programok nem profitálnak ebből a megoldásból.

A WDM Stíl Image architektúra operációs rendszer szintű támogatást nyújt lapelvasók és digitális kamerák alkalmazásához. Ennek révén a felhasználók egy egységes, integrált felületet kapnak a lapelvasók és foteleszközök egységes kezeléséhez. A Windows korábbi verzióiban a hardver fejlesztőjének kellett arról gondoskodnia, hogy az eszköz megfelelően együttműködjön az operációs rendszerrel.

### Enterprise Memory Architecture (EMA)

A memóriakiosztás sok nagy alkalmazáskiszolgálónál jelenthet szűk keresztmetszetet. Ez különösen igaz az akár több száz gigabájt adat kezeléséért felelős adatbáziskiszolgálóknál. A Windows 2000 Server egyik újítása az Enterprise Memory Architecture (EMA), melynek segítségével 32 bites processzorokkal akár 64 GB memória kezelése is lehetséges. A legtöbb kiszolgáló nem igényel ennyi memóriát, de bizonyos adatbázisokkal ez hasznos lehet, hiszen a memóriában lévő adatokkal sokkal gyorsabban lehet műveleteket végezni, mint a merevlemezen lévőekkel. Nem minden rendszer képes ennyi memória kezelésére, de az Alpha és Pentium II Xeon processzorok már ma is ki tudják ezt használni.

Az alkalmazásokat fel kell készíteni a VLM (*Very Large Memory*) API használatára. A Microsoft SQL Server erre már fel van készítve, és minden bizonnyal a többi relációs adatbáziskezelőből is készül majd ilyen verzió. A legtöbb alkalmazásnak azonban nincs szüksége 4 GB-nál nagyobb memóriára.

### Hatékonyabb többprocesszoros működés

Azon kiszolgálóknál, ahol a processzorteljesítmény jelenti a szűk keresztmetszetet, a Windows 2000 Server továbbfejlesztett többprocesszoros támogatása nagy előnyt jelenthet. Bár a Windows NT mindig is támogatta a többprocesszoros rendszereket, az új generációs kiszolgáló program ezt még hatékonyabbá teszi. Valamennyi többszálú alkalmazás gyorsabban fut többprocesszoros rendszeren, s ehhez nem kell alkalmazáskódot módosítani. A Windows 2000 Server négy processzor egyidejű használatát támogatja. Akik pedig Windows NT 4.0 Server Enterprise Edition rendszerükre cserélik Windows 2000 Advanced Server rendszerre, továbbra is legfeljebb nyolc processzort használ-

hatnak. A Windows 2000 Advanced Server megduplázza az eddig használható processorszámot – akár 8 processzor egyidejű használatát is támogatja. A hardvergyártóktól akár 32 processzoros rendszerek is rendelkezhetők.

Elsként a Windows operációs rendszerek közt, a folyamatokat egy kijelölt processzorhoz lehet rendelni. A rendszergazda a Task Manager segítségével beállíthatja a folyamatok processzoraffinitását. Ekkor a folyamat csak a kijelölt processzoron fog futni, ez növelheti a teljesítményt, mivel csökkenti a cache területek kiürítésének/feltöltésének számát, ami a processzorokon futó különböző folyamatok miatt szükségessé válhat. Óvatosan kell ezzel a lehetőséggel élni, hiszen akár a rendszer teljesítményének csökkenését is eredményezheti, mivel a folyamat ekkor nem kerülhet másik, esetleg kevésbé foglalt processzorra.

## I20 Támogatás

Az I20 (*Intelligens I/O architektúra*) egy új technológia, mely egyidejűleg csökkenti a processzor terheltségét, és növeli a rendszer I/O teljesítményét. Az I20 használatokor egy külön, dedikált processzor felel az I/O műveletekért. Ez különösen a nagy sávszélességűnél alkalmazásoknál hasznos (*valós idejű audio- és videóalkalmazások*).

## Rendezési alapritmusok

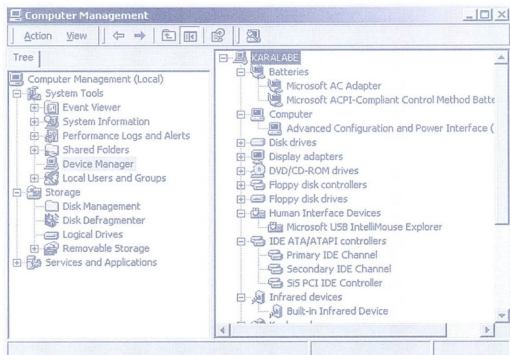
Nagy adatbázisrendszerek esetén lesz különösen hasznos a Windows 2000 Advanced Server továbbfejlesztett rendezési modellje. Ezek teljesítménye nagyban megnőtt, mivel az ezekhez szükséges processzorintézes algoritmusok most már kernel módban futnak. Az alkalmazásokat fel kell készíteni ezen új API használatára (*a Microsoft SQL Server legújabb verziója már támogatja ezt*).

## A Zero Administration for Windows támogatását szolgáló módosítások

A Plug and Play rendszer révén a rendszergazdáknak kevesebb időt kell tölteniük hardverek konfigurálásával és hibajavítással. Az Advanced Configuration and Power Interface (*ACPI*) hatékonyabb energiakihasználást tesz lehetővé, ezáltal csökkentve a fenntartási költséget. A lemezkvóták segítségével könnyebben kézben tartható a lemeztérület. A távoli rendszerindítás segítségével a munkaállomások telepítése jelentősen felgyorsul.

## Plug and Play

A számítógéphálózatok felügyeletének egyik legproblémásabb területe a hardverkonfigurálás. Régebben a hardver helyes beállításához elengedhetetlen volt a megszakítások, I/O portok és a Direct Memory Access (*DMA*) rendszer ismerete. A Plug and Play technológia segített ezen folyamatot egyszerűsíteni, de nem támogatta minden operációs rendszer. A Windows 95 és a Windows 98 teljes mértékben támogatta ezt a protokollt, de a Windows NT 4.0 csak minimális mértékben. A Windows 2000 Server jobb Eszközkezelővel és Hardvervezérléssel rendelkezik mint bármelyik korábbi Windows verzió. Ezek az eszközök megbízhatóan felismerik és feloldják a hardverkonfliktusokat. Ez a kiszolgáló rendelkezésre állási idejét is javítja, mivel a rendszergazdáknak nem kell hosszú időt eltöltenie az újonnan telepített hálózati kártya, modem vagy merevlemez beállításával.



## » Az eszközkvétel (*Device Manager*) segíti a rendszergazda munkáját a hardverbeállításoknál.

Amikor az operációs rendszer elindul, automatikusan felismeri az új hardvert, és amikor egy rendszergazda bejelentkezik, elindítja a Hardvervezérlést. A legtöbb konfliktus a rendszer automatikusan észreveszi és feloldja, azon esetekben pedig, amikor a rendszergazdáknak manuálisan kell beavatkozni, munkáját könnyen elérhető online súgó segíti.

## OnNow/ACPI

Az OnNow/ACPI technológia az energiafelhasználás hatékonyabb vezérlését segíti. Ez főként a hordozható gépek tulajdonságainak fontos, a kiszolgálóknál általában nem annyira fontos szempont, hogy a gép mennyi energiát fogyaszt. Ezek a szabványok be vannak építve a Windows Driver Model rendszerébe. Ezért mind a Windows 2000 Server, mind a Windows 2000 család többi tagja éppen úgy támogatja, mint a Windows 98.

## Lemezkvóták

A Windows új verziójával lépett színre az NTFS fájlrendszer új verziója is. Egyik legérdekesebb újítása a felhasználók által használható lemeztérület korlátozásának lehetősége, melyről e lap-számban részletes cikket is olvashat. Ez lehetővé tesz hogy a fájl- és webkiszolgálók rendszergazdái pontosan kézben tarthassák a felhasználók által tárolt fájlokat. Lehetőség van úgy beállítani a rendszert, hogy a felhasználók ne tölthessék meg a partíciót!

## Távoli Rendszerindítás

A Windows 2000 Server új távoli rendszerindítása egyszerűbbé teszi a munkaállomások bekapcsolásának irányítását. Míg a kiszolgálók általában folyamatosan üzemelnek, a munkaállomásokat hasznos lehet munkaidőn kívül kikapcsolni. A központi kiszolgálóról szabályzott rendszerindítás lehetősége a konfigurálást is egyszerűsíti, hiszen nem kell minden géphez egyesével odamenni. Természetesen a hálózati kártyákat és azok meghajtóit fel kell készíteni a távoli rendszerindítás kezelésére. Az új kártyák képesek az indítási kérelmek fogadására, akár a gép kikapcsolt állapotban is. A Microsoft dolgozik ezen funkció szabványosításán, és saját rendszereiben támogatja is.

## Fürtözés

A hálózatokat intenzíven használó szervezeteknél a magas rendelkezésre állás alapkövetelmény. A leállás a magas fenntartási költség egyik legfontosabb okozója. Ez számos különböző okból bekövetkezhet:

- Hardverhibák
- Alkalmazáshibák
- Konfigurálás miatti újraindítások
- Frissítések és javítások

A Windows 2000 Advanced Server fürtözési eljárása lehetővé teszi hogy két kiszolgáló egymást helyettesítse. Ha az egyik kiszolgáló leáll, a másik egy percen belül automatikusan átveszi a helyét. A figyelőszolgáltatások érzékelik a hibát, és a kéréseket a második kiszolgálóhoz irányítják.

Tervezett leállásoknál egy kiszolgáló átveheti a másik feladatait, amíg az nem elérhető. Ezen folyamat révén minimális lesz a javítások miatti leállás. Ez leginkább a rendszergazdák munkáját segíti, hiszen így a legújabb javítások és frissítések telepítéséhez nem kell megvárniuk a munkaidő végét! *(Nono. Azért az erőforrások átmozgatása nem vértelen folyamat! A nyitott kapcsolatok megszakadnak, az el nem mentett munka elvesz stb. – a szerk.)*

A fürtözés másik célja a terhelés minél egyenletesebb elosztása több gép közt. Bár a Microsoft fürtözési technológiájának elsődleges célja a rendelkezésre állás javítása, használható terhelés-kiegyenlítésre is. Például a fájl- és nyomtatómegosztások több rendszer közt lehetnek tükrözve. A kéréseket a rendszer automatikusan elosztja két vagy több rendszer közt, és az ügyfelek egy kiszolgáló leállása esetén automatikusan másikat keresnek a fürtben.

A fürtözés magába az operációs rendszerbe beépített szolgáltatás. A fürtöket egyszerű felügyelni, mivel az adminisztrációs eszközök segítségével több kiszolgáló beállítása végezhető el egyidőben. Sok külső megoldással ellentétben a Windows 2000 Advance Server nem igényel mindehhez drága vagy egyedi hardvert. A Windows 2000 Advance Server fürtözési képességét mindenféle speciális eszköz nélkül, azonnal használhatjuk. Egy fürtön belül akár különböző konfigurációjú gépek is lehetnek, például egy kevésbé drága gép is lehet biztonsági kiszolgáló. *(Azért egy NetAcademia fürttanfolyam nem árt, mielőtt belevetjük magunkat. – a szerk.)*

Számos további szolgáltatás is ismeri már a fürtözést:

- WINS (Windows Internet Name Service)
- DHCP (Dynamic Host Configuration Protocol)
- Dfs (Distributed File System)

## Terminálszolgáltatás

A Windows 2000 Server az első olyan Windows operációs rendszer, ami beépített terminálszolgáltatással rendelkezik. Ez lehetővé teszi, hogy az ügyfelek interaktív alkalmazásokat futtassanak egy távoli kiszolgálón. Az ügyfél rendszere fogadja a bemeneti adatokat és megjeleníti az eredményeket a felhasználó képernyőjén. Az ügyfél gépe és a kiszolgáló közt az adatok a hálózaton közlekednek. Minden feldolgozás a kiszolgálón történik.



• A Terminálszolgáltatás tehermentesíti a munkállomást, mivel az alkalmazások a kiszolgálón futnak

A vékony ügyfél (*thin client*) olyan munkaállomás, ami csak minimális számú alkatrészt tartalmaz. Ahelyett, hogy a munkaállomásokba gyors processzort, sok memóriát és nagy merevlemez vásárolnánk, a hálózati Terminálszolgáltatás végzi a munka zömét. A vékony ügyfél egyszerűen továbbítja a kéréseket a kiszolgálóhoz. A felhasználó továbbra is egy jól felszerelt munkaállomást érzékel. Az ügyfélgépek hardverkövetelménye minimális, a Terminálszolgáltatás bármilyen munkaállomással jól használhatóak.

A kliensekre vonatkozó alacsony követelmények különösen hasznosak lehetnek régebbi gépparkkal rendelkező vállalatok esetében. A meglévő Windows for Workgroups 3.11 és Windows 95 rendszerek az operációs rendszer cseréje nélkül képesek ügyfélként működni. Ennek segítségével egy korosabb munkaállomás is képessé válik a legújabb 32 bites alkalmazások használatára, ami növeli élettartamát. Természetesen a Windows 2000 Server tartalmazza a terminálügyfélprogram telepítőjét Windows NT 4.0, Windows 95, Windows 98, és Windows 2000 Professional rendszerekhez is.

Mivel a kiszolgáló végzi a feldolgozás nagy részét a hálózatban, a Terminálszolgáltatás erősebb hardvert igényel. Ez a többletköltség azonban megtérül az olcsóbb munkaállomások vásárlásakor. Ez az architektúra további költségsökkentő tényezőket is tartalmaz. Az alkalmazások magán a kiszolgálón helyezkednek el, így a rendszergazdának is kevesebb a munkájuk ha valamit módosítani kell. A felhasználói környezet is jobban kézben tartható, mivel teljes egészében a kiszolgálón található.

A Windows 2000 Server kernelének tervezését számos helyen befolyásolták a Terminálszolgáltatás szükségletei. Korábban a Windows operációs rendszerek egyetlen interaktív folyamat futtatására voltak felkészítve, azon folyamatára, amit a gép előtt ülő felhasználó futtatott. Annak érdekében, hogy a hálózaton keresztül futtatott több interaktív folyamat helyesen működjön, a Windows 2000 egy erősen módosított Win32 alrendszert használ. Ez az új alrendszer képes a különböző alkalmazásokat egymástól függetlenül követni és tárolni. A billentyűzet- és egéparancsok nemcsak az adott alkalmazásra, de a felhasználó teljes munkakörnyezetére vonatkoznak. Minden felhasználóra a saját biztonsági megszorításai érvényesek. Technikailag egy külön Win32 folyamat kezel minden felhasználói kapcsolatot. Ez biztosítja azt, hogy az alkalmazások nem kommunikálnak a felhasználói kapcsolatok közt, ami megakadályozza a felhasználói jogok megsértését. A kernel ezen módosításai nincsenek hatással azon rendszergazdák munkájára, akik nem kívánják használni a Terminálszolgáltatást.



### Globalizáció

Az eredeti Microsoft operációs rendszerek az Egyesült Államok vevőközösségének készültek. Csak a Latin ábécét tartalmazták, és az angol nyelvű szövegeket támogatták, és nem tartalmaztak olyan segédeszközöket, amik a nemzeti nyelvű alkalmazások támogatták volna. Mivel a Windows iránti igény a világ többi részén is felmerült, a fejlesztők más országok számára is készítettek alkalmazásokat. Ez a fejlesztés nehézkes volt, hiszen az operációs rendszer maga nem volt ilyen használatra felkészítve. Az újabb Windows verziók ezért egyre több módon támogatták a más nyelvű felhasználókat, és a nekik alkalmazást készítő fejlesztőket. A Windows 2000 operációs rendszer tartalmazza valamennyi Windows közt a legtöbb globalizációt segítő eszközt.

A Windows 2000 kernel speciálisan a globális igények kielégítésére lett felkészítve. Unicode karaktereket használ maga a rendszer is, ami nagyban egyszerűsíti a nemzeti verziók elkészítését. Az NLS, a Native Language Suport lehetővé teszi a helyi és nyelvi információk rögzítését a rendszerleíró adatbázisban, amire építve a felhasználói felület ennek megfelelő felületet kínál(*hat*). Ezek révén a fejlesztők nemzetközi környezetben is helyesen működő alkalmazásokat fejleszthetnek, a felhasználók pedig még jobban testre szabhatják munkakörnyezetüket.

### Unicode betűtípusok?

A Unicode majdnem 40,000 karaktert kódol. Nincsen olyan betűtípus, ami valamennyit képes lenne megjeleníteni. A különböző betűtípusok a Unicode karakterkészlet különböző részeit támogatják, így az egyes nyelvek megjelenítéséhez továbbra is helyi betűtípusok telepítése szükséges.

A Windows 2000 támogatja a Unicode karakterkészletet, és az operációs rendszer mindenhol ezt használja. Például valamennyi NTFS fájlnev Unicode karakterekből áll. Ez lehetővé teszi ugyanazon fájlrendszer használatát bármilyen nyelvi környezetben. Az alkalmazásokat fel lehet készíteni a Unicode által nyújtott előnyök kihasználására, de ezt nem minden alkalmazás igényli.

### Native Language Suport (NLS)

Az NLS lehetővé teszi, hogy a rendszergazda az operációs rendszert helyspecifikus információk figyelembe vételével állítsa be. Az API a Windows NT első verziótól kezdve az operációs rendszer része. Ez közel száz beállítás rögzítését jelenti a rendszerleíró adatbázisban, és biztosítja az ezekhez való hozzáférést, így az alkalmazások egyszerűen meghatározhatják az adott rendszeren érvényes helyi beállításokat. A helyspecifikus beállítások többet jelentenek a felhasználó által preferált nyelv rögzítésénél. A különböző országok különböző formában jelenítik meg a dátumokat, időket, a hét napjait, a pénzt, és még számos egyéb információt. Szabványos API-k használatával a felhasználónak nem szükséges minden egyes alkalmazásnál beállítania ezeket a jellemzőket. Az operációs rendszert egyszer kell beállítani, ezután minden alkalmazás lekérdezheti a rendszer beállításait.

Csak az adott nyelvhez és országhoz szükséges fájlok kerülnek telepítésre. Ezek helyigénye sokkal kisebb ahhoz képest, ha valamennyi a Windows 2000 által támogatott nyelvhez tartozó fájl telepítésre kerülne.

### Összegzés

A kernel a Windows NT Advanced Server 3.1 architektúrájához hasonló alapokra épül, a modern hálózati igények kielégítéséhez szükséges módosításokkal. A kernel révén az operációs rendszer jobban skálázható, 32 processzor és 64 GB memória használatát támogatja. Lehetővé teszi hogy adatbázisrendszerek, mint például a Microsoft SQL Server kihasználhassák az erősebb hardvert, és a sebességkritikus részek kernelbe integrálásával tovább növeli azok teljesítményét. A processzorkvóták és a folyamatszámolás a Windows 2000 Servert ideális rendszerré teszik webes kiszolgálószerepre. A most már kernelbe integrált fűrtözési lehetőség a redundáns hálózatok rendelkezésre állási idejét növeli. Végül, de nem utolsón sorban a Windows 2000 Server rendelkezik valamennyi Windows operációs rendszer közt a leg rugalmasabb nyelvi és nemzeti támogatással.



# Mixed vs. Native

Áttérünk, áttérünk. Windows 2000-re. Először a tartományvezérlőn, mert azokból kevés van. A munkaállomások ráérnek! Sok vállalat jutott már el a Windows 2000 áttérés azon szakaszába, amikor fontolóra veszik a tartomány átállítását Mixed módról Native módra. Ilyenkor szembesülnek azzal, hogy a két üzemmód közötti különbség eléggé homályosan dokumentált, s a helyzetet csak rontja, hogy különböző legendák és babonák keringenek a neten a Natív mód kegyetlenkedéseiről. Az izguló rendszergazda ezeken a kérdéseken rágódik hetekig:

- ☞ Lesz-e NetBIOS támogatás?
- ☞ Mi lesz a PDC emulátorral?
- ☞ Lesz-e továbbra is NTLM hitelesítés, vagy csak Kerberos marad?
- ☞ Képesek lesznek-e régi operációs rendszereink felvenni a kapcsolatot a Native módu tartománnyal?
- ☞ Fognak-e működni a régi típusú házirendek (NTCONFIG.POL)?

A fordítottjára is láttam példát, hisz töprengésre adhat okot az is, hogy mi VAN Mixed módban, hisz az nem „olyan jó” üzemmód:

- ☞ Vajon DNS-t használnak a Windows 2000 Pro kliensek a tartományvezérlő megtalálására?
- ☞ Van Kerberos?
- ☞ Megtalálják-e a munkaállomások, hogy melyik telephelyre tartoznak?

Hovatovább oda jutottunk, hogy a babonák nagyobb súllyal esnek latba a döntés meghozatalakor, mint a tudomány, s e közepkori gondolkodás miatt áttálatás a Native módra történő átállás el is marad. Pedig mi mindent nyernénk!

De kezdjük a történetet az üzemmódok elemzésével, s ez reményeim szerint eloszlat egy-két tévhitet.

## Kompatibilitás

A Windows 2000-re történő áttérés nem okozhatja a régi, bevezetett rendszerek működéséptelenné válását, hisz ha így történne, bizony senki még csak meg sem fontolná a frissítést. Mi mindenre kell gondolnia Redmondnak, ha vadonatúj operációs rendszernek kompatibilisnek kell maradnia a régi termékekkel! Futnia kell a régi szoftvereknek, a régi munkaállomásoknak csatlakozniuk kell az új tartományhoz, és – hogy a vállalati cím tár tartalma külön erőfeszítés nélkül átkerüljön az új rendszerbe - meg kell tudni upgradelni a tartományvezérlőket. Támogatni kell a vadonatúj protokollok (IPSec) mellett a régieket is (NetBIOS). Kemény feltételek ezek! Ezért is gondolják sokan, hogy ez csak ideig-óráig tartható, de előbb-utóbb eljön az igazság pillanata, s a régi vackokkal le kell számolni. Noha ez így igaz, alapvető tévedés azt hinni, hogy a Native módra történő átállás okozza ezt a kompatibilitási törést. Ugyanis a Mixed és Native üzemmódoknak **semmi közük** a végfelhasználói rendszerekkel történő kapcsolattartás kompatibilitási szintjéhez. Ha az előbbi kérdéseket megismételném (van-e NetBIOS, WINS stb. Native módban), az lenne rá a helyes válasz, hogy van bizony, hisz ezek ügyfélhozáférési technológiák, melyek elérhetők mindkét üzemmódban.

## Akkor hát mi a Native üzemmód?

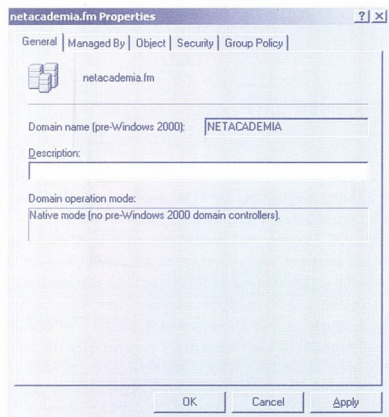
Van köze a kompatibilitáshoz, de más szinten. Ha összehasonlítjuk a Windows NT „cím tárát”, a SAM-ot és az Active Directory-t, látunk néhány lényeges különbséget. Az AD hierarchikus, míg a SAM kétdimenziós. Az AD-ban bővíthető sémájának köszönhetően tetszőleges objektum tárolható, a SAM-ban csak olyasmis, amit a SAMDRV.DLL lehetővé tesz: felhasználók, csoportok, számítógépek (Computer Account). Az AD-ban a replikáció alapegysége a pincy attribútum, a SAM-ban a user. Az AD-ban több százmillió objektum tárolható, a SAM-ban maximum negyvenezer. A SAM-okból csak egyetlen példány módosítható (a Primary Domain Controller példány írható/olvasható, míg a Backup Domain Controller csak olvasható), az Active Directory-nél mindegyik tartományvezérlő írható másolatot tárol a cím tár-ból. Az Active Directory lehetővé tenné, hogy azonos típusú csoportokat ágyazzunk egymásba (globálist globálisba), a SAM ezt nem viseli el. Az AD-ban létezik egy olyan csoporttípus, amely még a Global-nál is globálisabb, ez a Universal típusú csoport – no ez sem nyeli le a SAM. Olyan súlyos különbségek ezek, hogy szinte lehetlenné teszik a régi és az új együttélését. A szabadon származó Active Directory-t gúzsba kell kötni ahhoz, hogy régi típusú Backup Domain Controllerrel együtt tudjon működni. Na ezt hívják Mixed módnak.

## Mixed mód

Mixed módban lehetséges van arra, hogy egy tartományban vegyesen használjunk Active Directory tartományvezérlőket és NT4 BDC-eket (PDC-t nem mert az nem tűr meg még egy dudát a csár-dában, márpedig az Active Directory egy dudás, hiszen definíció szerint írható cím tár kópiával rendelkezik). Mixed módban az Active Directory csak olyan adatot fogad be a módosítások alkalmával, amit vagy nem is kell „lejelentenie” a BDC-knek, mert SAM nem ismeri őket (például cím tárba publikált nyomtatásobjektumok), vagy ha oda kell adnia a BDC-knek, akkor képes legyen rá. Azaz a hierarchia kisimítható legyen (ennek feltétele, hogy minden objektum egyedi SamAccountName tulajdonsággal rendelkezzen), ne legyen több, mint negyvenezer, ne legyenek Universal csoportok stb. A SAM replikációt egyébként a Windows 2000 tartományvezérlőn futó PDC emulátor végzi. A Mixed módnak **semmi köze** ahhoz, hogy a munkaállomások képesek-e a DNS segítségével bejelentkezni, telephelytárolások igazodni stb. Ez kizárólag a kliensgépek műik. Mixed módban is megteik a DNS zóna az Active Directory dinamikus bejegyzéseivel, melyekkel meg lehet találni a globális katalógus, mellyel meg lehet találni a telephelyek „szélet”.

## Native mód

A Mixed módra addig van szükség, amíg a tartomány vegyesen tartalmaz NT4 és Active Directory tartományvezérlőket. Amint az utolsó BDC-t is megupgradeltük, áttérhetünk Native módra, ami az eddigiek értelmében egy rabláncaitól megszabadított Active Directory-t ad a kezünkbe.



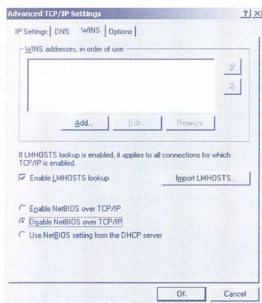
A falak leomlanak, a szürke menüpontok kigyúlnak, és szabadon garázdálkodhatunk:

- ☞ Lehet készíteni Security Universal csoportot
- ☞ Policyval szabályozható a RAS kapcsolat
- ☞ A csoportok típusa menet közben megváltoztatható (!). Unod a Local csoportot? Alakítsd át Globállá!
- ☞ A csoportok egymásba ágyazhatók. A tagja lehet B, melynek tagja ismét A. Ennek a levélküldésben vehetjük egyébként hasznát
- ☞ És ne feledkezzünk meg arról sem, hogy innettől több, mint negyven ezer objektumunk lehet (de kinek kell?)

S mi történik a felhasználói kapcsolatokkal? SEMMI. Ha eddig volt NetBIOS, ezután is lesz, hisz ennek semmi köze a címtár felszabadításához! Bizonyíték kell?

### Kompatibilitási kegyetlenkedések

Kezdjük a NetBIOS-sal, tiltsuk le! Ehhez a Start Menü->Settings->Network and Dial-up Connections->Local Area Connection->Properties->Internet Protocol (TCP/IP)->Properties->Advanced->Wins nevű elhagyott helyre kell ellátogatnunk, sötétedés után ne járjunk arrafelé magányosan, veszélyes környék ;)



☞ A NetBIOS támogatás leltitása

Ha itt átkapcsolunk Disable NetBIOS-ra, akkor valóban elveszítjük a régi klienseket (*beleértve az NT4-eket is*) de ennek semmi köze a Native módhoz, ugyanez a lehetőség ugyanis Mixed módban is kiválasztható. Vagy vegyük az NTLM hitelesítést: ha megszűnne, ismét csak elveszíténénk egy csomó munkaállomással a kapcsolatot. Kikapcsolható az NTLM? Persze! De ennek eszköze a csoportos házirend (*Group Policy*), és nem a Native módra történő átállítás!

### PDC Emulátor

S mi lesz a PDC Emulátorral? Megszűnik létezni? Nem! Nem szűnhet meg, mert feladata nemcsak a BDC-k felétörténő SAM-szerű replikációban volt, hanem fontos kompatibilitási szerepe is van: bizonyos kliensek állhatatosan keresik a PDC-t, még akkor is, ha ennek funkcionálisan semmi értelme sincs, és soha nem is volt: a Windows 9x-ek imádják a PDC-t, állandóan szólítgatják, pedig még a jelszóváltoztatást is beküldhetnek a BDC-nek, hisz az továbbítja a kérést a PDC felé. Ha pedig nem találunk PDC-t, nagyon bánatosak lesznek, és különböző IPCS nevű megosztásokra futnak rá, ahol persze semmi keresnivalóuk nincs. A PDC Emulátor szerep megmarad – hisz semmi köze a SAM tablánca veréséhez!

### NTCONFIG.POL

A régi típusú házirendek tökéletesen fognak működni, ha a házirendfájlt bemásoljuk a NETLOGON megosztásba.

### Fontos átállni Native módra?

Nem. Nincs olyan címtárfunkció, amely – nekem – hiányozna a Mixed üzemmódból. A séma Mixed módban is bővíthető, az Exchange 2000 feltelepíthető, a több forrású replikáció működik, a támogatott kliensek használhatják a Kerberos... mi kell még?

### Zárszó

Tudjuk, hogy a Native módra történő átállás veszélytelen (azt az egy veszélyt kivéve, hogy nincs visszaút). Ha valakit ezidáig nem sikerült meggyőzőm a Native mód jelentőségéről, most már nem is fog sikerülni. Remélem kiderült, hogy nincs szorító kényszer a lépés megtételére. Tüzetesen végigbongésztem a Resource Kit Deployment Planning Guide-ját, de semmi olyasmit nem találtam, ami mágnesként vonzana a Native mód felé – ha csak a RAS policyt nem számítom. Köszönöm, megvagyok egymásba ágyazható csoportok, tízmillió objektum és csoportátalkítás nélkül.

Fóti Marcell, MCSE, MCT, MCDBA, MZ/X  
marcell@netacademia.net



# Lemezkvóta a gyakorlatban

## Bevezetés

A rendszergazdáknak sokszor okoz fejtörést az fájlok mennyiségének intenzív növekedése a kiszolgálókon. Sok helyütt egyre égetőbb ez a probléma, hiszen az Internethozzáférés egyre általánosabbá válik, így a dolgozók nagy mennyiségben juthatnak hozzá képfájlokhoz, egyszerű játékokhoz stb., amelyeket előszeretettel tárolnak a hálózati meghajtókon. Kézenfekvő, de hosszú távon nem a legkifizetődőbb megoldás a kiszolgálók merevlemezegységeinek bővítése. Ennél sokkal nehezebben járható út, ha kizárólag adminisztratív úton, számítástechnikai szabállyal próbáljuk meg korlátozni a dolgozók gyűjtési szenvedélyét. Ez a módszer ugyan ingyenes, de csak akkor működőképes, ha a rendszergazdák a hatóság szerepét is felvállalják. Vagyis a számítástechnikusokra hárul a szabályzatba ütköző fájlok eltávolítása a merevlemezekről. A tapasztalatok azonban azt mutatják, hogy a kemény fellépés hatására „engedetlenségi mozgalom” alakul ki a dolgozók körében: minden lehetséges módszert megragadnak arra, hogy a kedvenc képeket megtarthassák. Megtörtént eset, hogy az egyik dolgozó a Kérlek ne töröld le! elnevezésű mappában tárolta az illegálisnak minősülő fájlokat. Nem mindenki hagyatkozik azonban a számítástechnikusok jó szívére. Mások inkább áttevezésével próbálják elrejtetni féltett fájlukat a kutató pillantások elől. Az így kialakuló harc sok időt és energiát vehet el a rendszergazdától. Ennek a módszernek az alkalmazhatósága és hatékonysága tehát erősen megkérdőjelezhető. Legelegánssabb megoldásnak a dolgozók által felhasználható lemezterület korlátozása tűnik. Így a számítástechnikusoknak nem kell a hatóság szerepét betölteniük, és a dolgozók maguk dönthetik el, hogy a rendelkezésre álló területen a munkához kapcsolódó vagy a szórakozást szolgáló fájlokat tárolják-e.

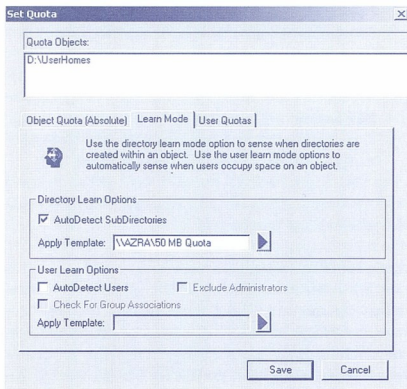
A Windows 2000 megjelenéséig a Microsoft operációs rendszerek nem tartalmaztak beépített lemezkvóta szolgáltatást. Számos programot fejlesztettek a Windows NT operációs rendszerre, amellyel korlátozni lehet a felhasználható lemezterület méretét. Ilyen például a W. Quinn Associates által készített QuotaAdvisor (4.1-es verzió) [1], a Northern Parklife féle Quota Server (5.1b verzió) [2] vagy az NTP Software Quota Sentinel (2.1-es verzió) [3] nevű alkalmazása.

## A konkurencia

Az ímént említett termékek közös jellemzője, hogy a Windows NT 4.0 mellett már a Windows 2000 operációs rendszeren is futnak. Mindháromra igaz az a megállapítás, hogy a szolgáltatások széles körével segítik a rendszergazdák munkáját. Jó példa erre, hogy a korlátozásokat nemcsak kötet-, de mappa- vagy akár fájlszinten is meg tudjuk adni. Ezen túlmenően felhasználókra, illetve felhasználói csoportokra is vonatkoztatjuk a kvótabeállítást. Ez utóbbi különösen olyan esetben hasznos, amikor például egy munkacsoportnak adunk hozzáférést jogosultságot egy hálózati meghajtóhoz. Az ímént említett szolgáltatás révén a csoport minden tagjának egysége-

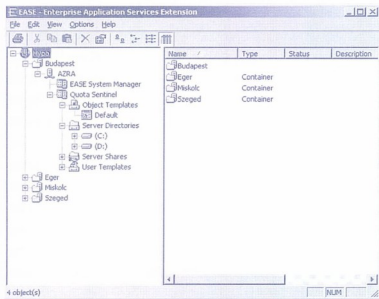
sen megszabhatjuk, hogy mekkora lemezterületet használhat fel az adott meghajtón. Ha új tag kerül a csoportba, akkor a kvóta természetesen rá is automatikusan vonatkozik.

A felsorolt három termék rendelkezik olyan beállítási lehetőséggel, amelyek eredményeképpen az újonnan létrejött objektumok automatikusan öröklődik a megfelelő kvótaértéket. Ezt a szolgáltatást a QuotaAdvisor esetében Learn Mode kvótának hívják, amely nagyon jól használható például a felhasználók kezdőmappájának (home directory) korlátozására. Ha arra a könyvtárra engedélyezzük ezt a beállítási formát, amely a kezdőmappákat tartalmazza (például \\Kiszolgálónév\ES\UserHomes), akkor minden újonnan létrehozott első szintű almappán (például \\Kiszolgálónév\DS\UserHomes\BurgundiM) érvényre jut az általunk meghatározott korlátozás. Felhasználókra is alkalmazhatjuk a kvótázás ezen típusát, amely nagyon hasznos a publikus mappák méretének szabályozásakor.



## ► QuotaAdvisor

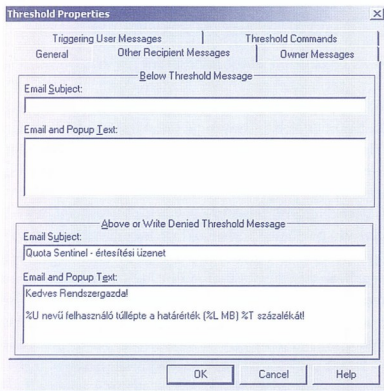
A három kvótázóprogram közül a QuotaSentinel 2.1-ben található a beállítások öröklődésének legösszetettebb formája. Az úgynevezett EASE technológia (Enterprise Application Services Extension) révén egységes felület áll rendelkezésünkre az összes Quota Sentinel kiszolgáló és az EASE technológiára felkészített alkalmazások kezelésére.



◀ **QuotaSentinel**

A kiszolgálók hierarchiába rendezésével leképezhetjük vállalatunk szervezeti felépítését. Az öröklődési szabályok értelmében a beállítások a hierarchia magasabb szintjétől az alacsonyabb felé terjednek tovább. Például ha az értesítési üzenetek küldését minden esetben ugyanarra a kiszolgálóra szeretnénk bízni, akkor ezt a hierarchia csúcán, a szervezet szintjén érdemes megállítani. A változtatások az adatok replikációja révén minden Quota Sentinel kiszolgálóhoz eljutnak.

Általánosan elterjedt szolgáltatásnak számít az értesítési üzenetek küldése is. A jellemzett alkalmazásoknál nekünk kell meghatározniuk, százalékos formában, azt a küszöbértéket, amelynek elérésekor a program figyelmeztető üzenetet küld. Például 50 MB-os kvóta és 80%-os küszöbérték esetén a program akkor küld értesítési üzenetet, amikor a felhasználó lemezerület mérete elérte vagy meghaladta a 40 MB-ot. Minden kvótabeállítás során lehetőségünk van több különböző küszöbérték meghatározására, melynek száma például a QuotaSentinel esetében nem kevesebb, mint 200! Az üzenetek létrehozásakor módunkban áll változókat (*küszöbérték, kvóta mértéke stb.*) is illeszteni a szövegbe, amely nagyban növeli a tájékoztatás információtartalmát.



◀ **A QuotaSentinel riasztási beállításai**

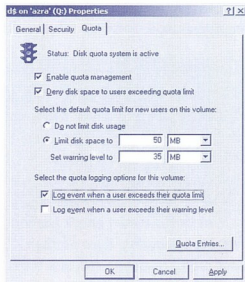
A figyelmeztető üzenet megjelenhet párbeszédablak formájában, amikor a felhasználó átlépte a határértéket, de érkezik elektronikus levélként is, például a rendszergazda címére. A QuotaAdvisor, a Quota Server és a QuotaSentinel is támogatja az SMTP protokollt.

Mivel az alkalmazások sokféle beállítási módot tesznek lehetővé, a lemezerület kihasználtságán nyomon követése nem könnyű feladat. Gondoltak erre a programfejlesztők is, így a kimutatások készítésére beépített eszközök állnak a rendelkezésünkre. Bár ezen eszközök által nyújtott szolgáltatások között nagy különbség mutatkozik az egyes kvótázó programok vonatkozásában, az aktuális állapot áttekintésében jelentős segítséget nyújtanak. Az elkészített kimutatásokat többféle fájlformátumban is elmenthetjük (*például text vagy HTML*), sőt a QuotaServer ODBC kompatibilis adatbáziskezelővel is képes együttműködni.

A kvótázóprogramok által nyújtott szolgáltatások teljes tárházának áttekintése túlmutat ennek a cikknek a keretein. Ezért e rövid ízelítő követően nézzük meg, hogy a Windows 2000 operációs rendszer mit kínál számunkra ezen a területen.

**Lemezkvóta a Windows 2000 operációs rendszerben**

A lemezkvótaszolgáltatás csak a Windows 2000 új fájlrendszere alatt használható. Nem alkalmazhatjuk tehát a lemezerület korlátozásának ezt a fajtáját például FAT-es meghajtókon. (*Az NT 4.0 NTFS fájlrendszerét a Windows 2000 automatikusan frissíti.*) A kvótázás csak kötetek (*partíciók*) szintjén engedélyezhető. Ezt azért fontos hangsúlyozni, mert egy lemezegységen több kötet is létrehozható, illetve egynél több fizikai lemez is állhat a kötet hátterében. Ebből az is következik, hogy sem a mappák, sem a fájlok szintjén nem tudjuk a korlátozást érvényesíteni. Ugyancsak fontos jellegzetesség, hogy a lemezerület felhasználásának ellenőrzése azon alapul, hogy ki a fájlok tulajdonosa. Csoportokra tehát nem szabhatunk ki lemezkvótát, hisz minden fájl egyetlen tulajdonossal bír – alapértelmezésben az a tulajdonos, aki létrehozta. Ez alól csak az Administrators csoport képezhetne kivételt, hiszen a tagjai által létrehozott fájlok tulajdonosa mindig a csoport lesz. Természetesen a rendszergazdák számára nincs értelme kvótát beállítani, sőt esetükben az operációs rendszer nem is teszi lehetővé a lemezhasználat korlátozását. A lemezkvótát a kötet Properties párbeszéd ablakban tudjuk engedélyezni. Ehhez a Quota fülön található Enable quota management jelölőnégyzetet kell kiválasztanunk.



◀ **A Windows 2000 kvótarendszere**

Első lépésként az operációs rendszer feltérképezi, hogy melyik felhasználó mekkora területet foglal el a szóban forgó kötetben. Alapértelmezésben az operációs rendszer kvótatüllépés esetén nem tiltja le a lemezterületet. A tiltáshoz a **Deny disk space to users exceeding quota limit** jelölőnégyzetet kell kiválasztanunk. Kétféle határértéket tudunk beállítani. Egyrészt megszabhatjuk, hogy a kötetben mekkora helytel gazdálkodhatnak a felhasználók (**Limit disk space to mezo**), másrészt lehetőségünk van figyelmeztetési értéket is megadni (**Set warning level to mezo**). Fontos felhívni a figyelmet arra, hogy az itt megadott értékek akkor van szerepük, amikor az operációs rendszer a kvótát hozzárendeli a felhasználókhöz. Vagyis a kvótázás legelső engedélyezésekor, illetve akkor, amikor egy új felhasználó hoz létre (*vagy vesz tulajdonába*) egy fájlt a szóban forgó kötetben. A **Limit disk space to**, valamint a **Set warning level to mezo**k tartalma tehát nincs hatással a már érvényben lévő kvótabeállításokra.

A határértékek tüllépéséről bejegyzés készül az eseménynaplóba, amennyiben a **Log event when a user exceeds their quota limit**, valamint **Log event when a user exceeds their warning level** beállítást engedélyezzük.

Az események naplózására alapértelmezésben óránként kerül sor. Ezt az időintervallumot a regisztrációs adatbázisban tudjuk módosítani. A

```

HKY_LOCAL_MACHINE\System\CurrentControlSet\
Control\FileSystem
    
```

kulcs alatt hozzuk létre a **NtfsQuotaNotifyRate (DWORD)** nevű bejegyzést, amelynek értékét másodpercekben számolva kell meghatározniuk. Sajnos a figyelmeztetési érték tüllépéséről az érintett felhasználót az operációs rendszer semmilyen formában sem értesíti.

A kvótabejegyzéseket a Quota fülön található Quota Entries gomb megnyomásával tudjuk megtekinteni.

Status	Name	Logon Name	Amount Used	Quota Limit	Warning Level	Percent Used
OK	BULL TPJA\Administrators		246,48 MB	No Limit	No Limit	N/A
Warning	Károlyi Rózsika	KarolyiR@lppcs.hu	45,91 MB	50 MB	35 MB	91
OK	Melkon Wilsoo	MelkonW@lppcs.hu	34,15 MB	50 MB	35 MB	68
Warning	Ronda István	RondaI@lppcs.hu	38,94 MB	50 MB	35 MB	77

### Windows 2000 kvótabejegyzések

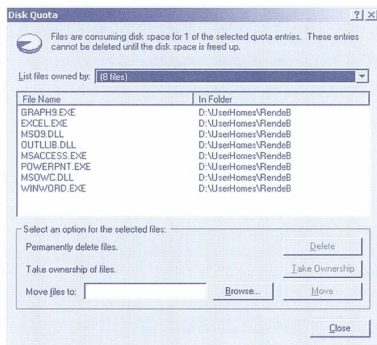
Mint a képen is látható, az Administrators csoport tagjai számára alapértelmezésben nincs beállított határérték. Ahogy korábban már volt róla szó, kvótát nem is tudunk a csoport számára meghatározni, csak a figyelmeztetési értéket tudjuk megváltoztatni. A párbeszédablakról a kvótabeállításokkal kapcsolatos minden fontosabb információ leolvasható. Az áttekinthetőséget növeli, hogy az adatok az egyes oszlopok szerint sorba rendezhetőek. Több száz felhasználó esetén ennek ellenére sem könnyű megtalálni a keresett bejegyzést. Ilyen esetben az **Edít** menü **Find** parancsa lehet segítségünkre, amelyel a teljes bejelentkezési név alapján tudunk keresni. Sajnos az első néhány karakter begépelése nem elegendő, ekkor érdemes lehet más elemzőeszközt (*például Excel-t*) igénybe venni.

### Nagyon izgalmas lehetőség a fenti ablak drag-and-drop képessége: egyszerűen húzzuk át Excelbe a bejegyzéseket!

Az egyes dolgozókhöz tartozó határértékek is ezen a fülleten lehet megváltoztatni. Egy időben nemcsak egy bejegyzés tulajdonságát tudjuk megjeleníteni, így a több felhasználót érintő változtatás egyszerűen is megvalósítható.

A Quota menü **New Quota Entry** parancsával olyan felhasználók számára is létrehozhatunk bejegyzéseket, akiknek nincs birtokukban fájl a szóban forgó kötetben. Ez a lehetőség akkor lehet hasznos, ha nem vezetünk be általános érvényű korlátozást (*a Quota fülön a Do not limit disk usage rádiógombot jelöljük ki*), csak néhány dolgozó számára akarjuk megszabni a felhasználható lemezterület méretét.

A felhasználói azonosító törlésével a kvótabejegyzés nem törlődik automatikusan. Mivel az operációs rendszer már nem tudja a bejegyzést egy létező felhasználói objektumhoz társítani, ezért a Quota Entries ablak Name oszlopában az [Account Information Unavailable] felírat, míg a Logon Name oszlopban a felhasználó SID-je jelenik meg. A kvótabejegyzés eltávolítását azonban mindaddig nem végezhetjük el, amíg a kötetben található olyan fájl, amelynek a törölt felhasználó a tulajdonosa. Erről az alábbi párbeszédablak tájékoztat:



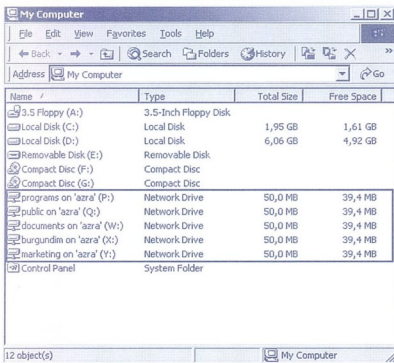
### Mi történjen a törölt felhasználó fájljaival?

Több választási lehetőségünk is adódik. Egyrészt saját tulajdonba vehetjük a szóban forgó fájlokat (*Take Ownership nyomógomb*). Másrészt átmozgathatjuk őket egy másik kötetre (*Move files to mezo*). Harmadrészt véglegesen letörölhetjük ezeket a fájlokat (*Delete nyomógomb*).

A rendszergazdai feladatok egyszerűsítése végett lehetőségünk van a felhasználókhöz tartozó kvótabeállításokat átvinni egyik kötetről a másikra. Ezt legegyszerűbben úgy tehetjük meg, hogy megjelenítjük mindkét kötet Quota Entries ablakát, majd a forráskötet ablakából egyszerűen áthúzzuk a kijelölt kvótarekordokat a másik Quota Entries ablakba. Arra is lehetőségünk van, hogy a kijelölt felhasználók beállításait Quota menü **Export** parancsával egy tetszőleges kiterjesztésű fájlba elmentjük. Szükség esetén ezeket az adatokat bármelyik kötetben érvényre juttathatjuk a Quota menü **Import** utasításával. A kvótabeállításokat Windows 2000 Professional operációs rendszeren futó munkaadomáról is kezelhetjük. Ehhez a megosztott kötetet egy meghajtójelhez kell hozzárendelnünk.

## Lemezkvóta a felhasználók oldaláról

A felhasználók számára az egyik legfontosabb kérdés, hogy mennyi szabad helyel rendelkeznek a hálózati meghajtókon. A munkavégzés szempontjából nagyon lényeges, hogy mennyire könnyű ezzel kapcsolatban pontos információhoz jutni. A szabad lemezterület nagyságát a legegyszerűbben a My Computer ablakban tekinthetjük meg. Ha engedélyezve van a lemezkvóta, akkor a hálózati meghajtó teljes kapacitásként a felhasználóhoz rendelt kvótáérték jelenik meg. Természetesen az is egyértelműen megállapítható, hogy mennyi a felhasználható szabad terület. A bonyolalom akkor kezdődik, ha ugyanazon a kvótázott kötetben több különböző megosztáshoz is csatlakozott a felhasználó. Ekkor válik ugyanis nyilvánvalóvá, hogy az operációs rendszer a kötetben lévő szabad terület méretét mutatja meg.

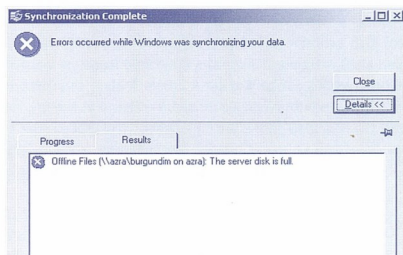


☛ **A hálózati megosztások ugyanazon a kvótázott kötetben találhatóak. A My Computerben megjelenő adatok azt sugallják, hogy mindegyik hálózati meghajtón majdnem 40 MB üres hely van, holott összesen ennyi a felhasználható szabad lemezterület**

Ez félrevezető, főleg, ha figyelembe vesszük: a felhasználó nem látja, hogy az adott hálózati megosztás a kiszolgáló lemezegységének melyik kötetben található. A fájlkezeléssel kapcsolatban is megfigyelhető némi bizonytalanság. Ha ugyanazon a kvótázott kötetben két megosztáshoz kapcsolódunk, és megpróbálunk a rendelkezésünkre álló szabad helynél nagyobb méretű fájlt átmozgatni az egyik hálózati meghajtóról a másikra, akkor „Nincs elég szabad lemezterület” hibáüzenetet kapunk. Ez egyrészt azért meglepő, mert a mozgatókór nem nő a lefoglalt lemezterület mérete. (Network Monitorral persze látszana az igazság, ennek fényében a jelenség már egészen érthető lenne – szerk.) Másrészt azért is különös, mert ha ugyanazon a hálózati meghajtón az egyik mappából a másikba mozgatjuk át a fájlt, akkor az operációs rendszer gond nélkül elvégzi a feladatot. A felhasználók számára az elosztott fájlrendszeren (Distributed File System - DFS) keresztül is biztosíthatunk hozzáférést a hálózati erőforrásokhoz. A DFS segítségével a hálózaton fizikailag elosztott megosztásokat úgy jeleníthetjük meg, mint ha a hálózat ugyanazon helyén lennének. Ez a gyakorlatban azt jelenti, hogy a felhasználó csak egy hálózati meghajtót lát (DFS-gyökér), és a különböző kiszolgálókon lévő megosztások

úgy látszanak, mintha a hálózati meghajtón lévő mappák lennének. Abban az esetben, ha a DFS-gyökér kvótázott kötetben található, akkor a felhasználó a hálózati meghajtó kapacitásként a neki beállított kvótáértéket fogja látni. Ez abban az esetben fedi a valóságot, ha a DFS-gyökér és a hozzá kapcsolott megosztások ugyanazon a kötetben találhatóak. Ellenben, ha a hálózati megosztások különböző kvótázott kötetben helyezkednek el, akkor a felhasználó számára gyakorlatilag követhe-telenné válik, hogy hol mennyi szabad hely áll a rendelkezésére. Hiába jeleníti meg ugyanis a DFS-gyökér alatti mappák tulajdonságait, semmi erre vonatkozó adatot nem fog találni. Szintén lényeges kérdés, hogy az alkalmazások milyen hibáüzenetet küldenek, amikor a felhasználó már kihasználta a rendelkezésére álló szabad területet. A bonyolult, nehezen érthető, hexadecimális hibakódokkal tüzeltet üzenet ugyanis komoly aggodalmat válthat ki a dolgozókból, amelyet csak a rendszergazda megnyugtató szavai csillapíthatnak. Ebből a szempontból a Windows 2000 Professional és a Windows NT 4.0 Workstation „beépített” alkalmazásai (Wordpad, Windows Explorer stb.) eredményesen vonatgáltak. Az üzenetek rövidek, jól értelmezhetőek voltak, és a tájékoztató valamilyen formában szerepelt a disk full vagy a not enough space on the disk kifejezés. Egyedül a Windows 2000 Notepad-je küldött némileg megtévesztő hibáüzenetet. Abban az esetben, ha a kötetben már létezik a text fájl, amit szerkesztünk, de elmentésekor már túllépünk a határértéket, a Notepad a következő tájékoztatást adja: Cannot open the file. Make sure a disk in the drive you specified. Érdekes, hogy ezzel szemben a Windows NT 4.0-ban található Notepad ebben a speciális esetben is korrekt tájékoztatást küld.

Hasonlóan jók a tapasztalatok az Office 97 és az Office 2000 alkalmazásai esetében. Az operációs rendszer még a makróval elvégzett fájlműveleteknél is egyértelmű visszajelzést adott. A Windows 2000 Professional-ban újításként jelent meg az Offline mappák használatának lehetősége. A szolgáltatás lényege a következő: bármelyik hálózati mappánál megadhatjuk, hogy a könyvtárban szereplő fájlok akkor is elérhetőek legyenek a munkaadómáson, amikor nincs hálózati kapcsolat. Amint helyreáll az összeköttetés a kiszolgálóval, az operációs rendszer szinkronizálja a kiszolgálón lévő példányt a helyi másolattal. Előfordulhat azonban olyan eset, hogy a lokális fájl mérete annyival megnövekedett, hogy a szinkronizálás során túllépünk a kvótát. Ilyenkor az operációs rendszer nem tudja frissíteni a hálózati meghajtón lévő példányt. A tájékoztató üzenetben egyértelműen szerepel a kudarc oka: a kiszolgáló lemezegysége megtelt.



☛ **A kvóta igazságos: az Offline dolgozókra is érvényes**



Az operációs rendszer továbbra is érzékeli, hogy a helyi és a hálózati adatok eltérnek egymástól, így a szinkronizálást megpróbálja elvégezni minden be- és kijelentkezéskor (*amennyiben a beállításnál engedélyztük ezt a szolgáltatást*). Speciális esetnek számít az eltérő adatfolyam (data stream) használata fájlok esetében. Az NTFS fájlrendszeren lehetőségünk van ilyen módon adatokat kapcsolni az fájlokhoz. Az adatfolyam a következő jellegzetességekkel rendelkezik:

- ☞ Nem jelenik meg sem az Explorer felületén, sem a parancssori DIR utasítás hatására.
- ☞ Tartalmának megjelenítéséhez ismerni kell a pontos nevét.
- ☞ Az operációs rendszerben nincs beépített eszköz arra, hogy a lemezegységen lévő adatfolyamokat kilistázzuk vagy letöröljük. Az előbbi feladatra nagyon jól használható a Frank Heyne LDAS (*List Alternate Data Stream*) [4] nevű ingyenesen letölthető programja.
- ☞ Létrehozásához nem szükséges kiemelt jogosultság, így a felhasználóknak lehetőségük van arra, hogy adatokat rejtessenek el a rendszergazdák előtt.
- ☞ Létrehozása többféleképpen is történhet. Például ha a teszt.txt nevű fájlhoz szeretnénk adatfolyamot kapcsolni, akkor parancssorból adjuk ki a következő utasítást: `notepad.exe teszt.txt:adatfolyam.txt`. Ennek hatására egy új, rejtett fájl jön létre, amely az elmentést követően szoros kapcsolatban lesz a `teszt.txt`-vel.
- ☞ A fájl másolása, mozgatása érinti a hozzá tartozó adatfolyamot is.
- ☞ Az adatfolyam eltávolítása csak közvetett módszerekkel oldható meg. Például ha a `teszt.txt`-t átmozgatjuk FAT-es partícióra, akkor az adatfolyam automatikusan törlődik, mivel ez a fájlrendszer nem támogatja az adatfolyam létrehozását. Ezen kívül a `teszt.txt` tartalmát átmásolhatjuk egy másik fájlba. Ezután a `teszt.txt` letörölésével megsemmisítjük az adatfolyamot is.

Mivel az adatfolyamok kezelésére nincsenek beépített eszközök, ezért felmerülhet a kérdés, hogy a Windows 2000 lemezkvóta szolgáltatása megkerülhető-e ezzel a módszerrel? A válasz egyértelműen, és hacker szempontból sajnálatosan: nem. A kvótát meghaladó mértékű adatfolyamot nem lehet elmenteni a lemezegységre. Más a helyzet azonban a cikkben szereplő kvótázó programok esetében. A tapasztalatok azt mutatják, hogy a QuotaAdvisor 4.1, a Quota Server 5.1b és a Quota Sentinel 2.1 alkalmazások közül egyedül a Quota Server 5.1b-t készítették fel az adatfolyamok kezelésére. A QuotaAdvisor 4.1 és a Quota Sentinel 2.1 esetében tehát kijátszható a korlátozás.

## Összegzés

A Windows 2000 operációs rendszer beépített lemezkvótája a szolgáltatások területén ugyan elmarad a piacon kapható komolyabb kvótázó programoktól, de megbízható, jól használható eszköz. Nagy hátránya viszont, hogy nincs lehetőség arra, hogy a felhasználók üzenetet kapjanak: mikor közelítenek a rendelkezésükre álló lemezterület felső határához. Ennek hiányában csak arról értesülnek, hogy nincs már több szabad hely például a dokumentum elmentésére.

Szintén problémák forrása lehet a lemezkvóta azon jellegzetessége, hogy csak felhasználók szintjén lehet a korlátozást érvényre juttatni. Ez főként a közepes és nagyvállalatoknál gyakori publikus meghajtók esetében jelenthet gondot. Jelleghénel fogva az ilyen hálózati meghajtókon a dolgozók egymás dokumentumait is szerkeszteni tudják. A fájl mentésekor azonban nem változik meg a fájl tulajdonosa. Ez alapjául szolgálhat a rosszindulatú tréfáknak: például igen nagy méretű kép illesztése a kevésbé kedvelt munkatárs tulajdonában lévő dokumentumba... Mivel sem a kötetekre, sem a könyvtárrakra nem szabhatunk ki kvótát, a dolgozók növekvő száma esetén nehéz biztosítani, hogy a lemezegység ne teljen meg. A tapasztalatok szerint érdemes arra is figyelni, hogy a felhasználónak ne legyen két különböző hálózati meghajtója ugyanazon kvótázott kötetről. Ez azért lehet fontos, mert ellenkező esetben nehezebb megállapítani a szabad lemezterület méretét. Ezen kívül a hálózati meghajtók közötti fájlmozgatás is komplikációkat okozhat.

Mindent egybevéve nagy előrelépést jelent, hogy a lemezterület korlátozását már az operációs rendszer szintjén is meg tudjuk oldani. A Windows 2000 ezen szolgáltatása számos esetben kielégítheti az igényeket. Valószínűleg azonban a konkurens kvótázóprogramok a közeli jövőben nem fognak eltűnni a piacról.

**Tomasz Balázs**

*balazs.tomasz@hu.hypovereinsbank.com  
pszichológus*

### A cikkben szereplő URL-ek:

- [1] <http://www.wquinn.com>
- [2] <http://www.northernparklife.com>
- [3] <http://www.ntpssoftware.com>
- [4] <http://www.heysoft.de>





# Szerszámoszláda (I. rész)

## Hasznos apróságok

Cikkorozatunkban a Windows 2000-hez kapcsolódó hasznos Microsoft segédprogramokat részletezem. Többségük parancsoros, bár jónéhány grafikus felülettel rendelkezik, és általában batch állományokból, scriptekből jól vezérelhetők. A vegyesfelvágott jelleget erősítő az alábbi forrásokból emelek ki alkalmanként néhány fontosabb eszközt:

- ☞ 1. Windows 2000 Server
- ☞ 2. Windows 2000 Support Tools
- ☞ 3. Windows 2000 Server Resource Kit

Először is tisztázom a különböző források elérhetőségét. Az első csoportba a Windows 2000 Server telepítések a rendszerben alapértelmezés szerint megtalálható programok kerültek. Ezen sorozatban kizárólag azokat a parancsokat szedem elő, amelyek messze értékükön és lehetőségeiken alul használnak, vagy általánosan nem ismertek. A Windows 2000 Support Tools tartalmához a már említett Windows 2000 telepítő CD-n a X:\SUPPORT\TOOLS könyvtár tartalmának telepítésével juthatunk hozzá. Itt a setup.exe lefuttatásával települ a support.cab állomány, teljességében kb. 20MB tárgyennyel. Sokan nem ismerik ezen ingyenes eszközkészlet hasznosságát, remélem sikerül majd bebizonyítanom létjogosultságát és fontosságát. A harmadik csoportba a *(sajnos)* fizetős Resource Kit CD mellékletén található segédprogramok tartoznak. Fontos megjegyezni, hogy ezen MS Press kiadvány *(és a Supplement One kiegészítésének)* beszerzése mellett más lehetőségeink is vannak. Lehet, hogy már rendelkezik ezen erőforrásokkal, csak még nem tud róla! A Microsoft Technet *(nem az újság, hanem a CD!!)* előfizetéssel járó CD-gyűjtemény tartalmazza a Resource Kit CD-k anyagát, valamint az MSDN előfizetők *(ezáltal a Certified Partnerek is természetesen)* az MSDN letöltési szkeciókból [1] érhetik el. Ezen kívül a nyilvános [2]-es címűl ingyenes minták is letölthetők. Természetesen az utóbbi opciók nem tartalmazzák a Resource Kit nyomtatott, méretes polcfoglaló anyagát, csak Windows Help formában kiadványokat. Ne feledjük, hogy a segédprogramok csak egy részét képezik ennek az értékes kiadványnak, és nem csak a Microsoft által készített kiegészítéseket tartalmazza.

A 80-100 MB tárhelyet foglaló anyag bőven ad lehetőséget a csemegézésre. A telepítés után azonban a CD-re még szükség lehet, az alkalmazások egy része manuálisan, később telepítendő onnan. A Supplement One feltételezi az alap Resource Kit meglétét, azonban főleg kiegészíti azt új, előre elkészített scriptek tömkelegével.

## Kategóriák

A Support Tools és a Resource Kit segédprogramjait kategóriákba rendezve találjuk meg .cab formában a CD-n, és telepítésük után is így értelmezhetjük őket legegyszerűbben. A Computer Management Tools között találhatjuk a processzerekkel, szolgáltatásokkal, a regisztrációs adatbázissal, az eseménynaplóval, és a felhasználói profilokkal foglalkozó programokat. A Deployment Tools szekció a tömeges, automatizált, kibővített telepítés eszközeiről szól. Itt összefuthatunk a rendszerházi rendek, nyomtatók és címtárak migrációjához, átmozgatásához nélkülözhetetlen eszközkészlettel és dokumentációval. A Desktop Tools kategória kizárólag a Resource Kit-ben van jelen. Többnyire grafikus kezelőfelületek, és a képernyő, a billentyűzet valamint a Windows Explorer kényelmét növelik. A Diagnostic Tools elnevezés magáért beszél, Isten adja hogy ne legyen rájuk szükségünk! Hálózati (RPC, PPTP, SNMP) és helyi eszközök itt vegyesen vannak jelen, a hardverközel és magasszintű, Active Directoryt és tartományvezérlőket diagnosztizálóok egyaránt ide tartoznak. A File and Disk Tools az alacsony szintű merevlemez-szerkesztéstől és állomány-összehasonlítótól, a vágólap és az állomány-kiterjesztések kezelésétől a DFS és Remote Storage ellenőrzéséig terjed. A Resource Kit IIS kategóriája a webkiszolgálók diagnosztikai és rendszerkezelési segédletét erősíti. A Network Management Tools lakosai az ADSI, DHCP, DNS, LDAP, AD; terminál-szolgáltatásért folytatott küzdelmekben jöhet jól. A Performance Tools alatt a teljesítmény-finomhangoláshoz elengedhetetlen győngyszemek rejtőzködnek. A scriptelésről szóló külön cikkek miatt nem részleteztem most a Scripting Tools tartalmát, de...

## Segítség

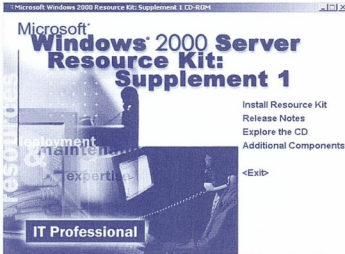
A dokumentáció erőforrásait függően változik, a Support Tools és a Resource Kit részletes, szabályos Windows Help formátumú anyaggal támogatja a tartalmát, amiket a Tools Help hivatkozással tudunk elérni. Az alap Windows 2000 parancsokat a gépen és a weben található sűgőből érthetjük meg. Ahol ez nem egyértelmű, megadom a szükséges webcímet. Az első alkalommal a „szerelem első látásra” kategória győzteseit választottam ki, különböző területekbe belekóstolva.

## Az IIS újraindítása: írsreset.exe

Windows 2000 Server része

x:\winnt\system32\iisreset.exe

Az írsreset segítségével az IIS szolgáltatásokat immáron meg-



☞ Szupergyűjtemény, első kiegészítés

bizhatóan tudjuk újraindítani karbantartási céllal, az alap operációs rendszertől függetlenül. Az iisreset parancssoros funkcionálitása nagyjából megegyezik az MMC-modul Internet Services Managerből elérhető „Restart IIS...” parancsral. Azonban a paramérezhetőséget figyelembe véve sokkal finomabban vezérelhető, és természetesen időzíthető is. Ez kifejezetten hasznos lehet a bizalmatlan (memória-telítődéstől tartó) rendszerfelújítás számára. Mellesleg az IIS dokumentáció [3] külön felhívja a figyelmünket arra, hogy ne az „alap” Services felülettel vezéreljük az IIS-t. Fontosabb paramérezése a következő lehetőségeket adja:

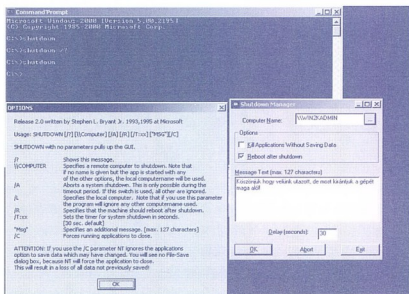
iisreset [gép neve] [paraméterek]

Ha nem adjuk meg a gép nevét, akkor alapértelmezés szerint a helyi szolgáltatásokra vonatkoztatja az utasítást. Paraméterek nélkül egy teljes körű szolgáltatás-újraindítást végez el. A szolgáltatásokat a /START és /STOP paraméterekkel indíthatjuk és állíthatjuk meg. A /RESTART az előző két paraméter kombinálja. A gépet a /REBOOT esetben indítja újra. A /REBOOTONERROR lehetővé teszi hogy kizárólag a szolgáltatások újraindítása során bekövetkező kritikus hiba esetén induljon újra a gép. A /NOFORCE kapcsoló erőszak-mentes viselkedésre készíti az iisreset-et, így nem próbálja végtelen időig elvégezni a feladatát. Az IIS újraindítását a /DISABLE és /ENABLE kapcsolókkal letilthatjuk és engedélyezhetjük. Egy ötlet a felhasználására: a kiszolgáló Services menedzserfelületén az IIS Admin Service esetében a Recovery fülön megadhatjuk (mint minden más szolgáltatás esetében is), hogy az adott szolgáltatás elhalálozása esetében lefuttasson egy parancsot. Ez a parancs lehet például az iisreset.exe (hiszen emlékszünk, hogy külön az IIS Admin Service újraindítása nem szerencsés).

**Remote Shutdown: távoli rendszerleállítás**

Windows 2000 Resource Kit  
Network Management Tools  
Shutdown.exe

Ez a kombinált parancssoros és grafikus felületű segédprogram szégyen felhasználók rémálma, és egyben a rendszerüzemeltetők mennyei segédlete. Adminisztrátori jogokkal rendelkezve segítségével pillanatok alatt újraindíthatjuk vagy lekapcsolhatjuk bármelyik munkaállomást a LAN-on. Ha paraméterek nélkül indítjuk, akkor a GUI jelenik meg, egyébként csendben konzolon is elfutkározik. A /? kapcsolóval kivallathatjuk közvetlenül a lehetőségeinkről, első közt a \\GÉPNÉV megadásával próbálkozzunk. A gonosz program még erőszakos, mentési lehetőség nélkül újraindítás is támogat, a /C paraméterrel. Saját tapasztalat, hogy a lekapcsolás nem teljesen kompatibilis az ACPI/ATX alapú Windows 2000 gépekkel, ezért a szokásos búcsúképernyő kiint marad, és a tápot sem kapcsolja le. Feltehetőleg a program NT4-es múltjában van a bibi. A /R kapcsoló újraindítás eredményez a lekapcsolás helyett, ez működik. A /T:xx használatával másodpercben megadható a figyelmeztetés és a kikapcsolás/újraindítás közötti idő. Ez alapértelmezés szerint 30 másodperc. A kedvenc funkció azonban az üzenet-beszúrás lehetőség („üzenet”), amivel is megfelelő haiku költeményeinkkel az öngyilkosságba kergethetjük felhasználóinkat. Talán csak a csoportos gépkiválasztás hiányzik a repertoárjából. A shutdown.exe természetesen időzíthető az AT vagy a Task Scheduler segítségével.



👁 **Kérjük vigyázzanak, az ajtók záródnak!**

**RDPCLIP - állományműveletek terminál-klienssel**  
Windows 2000 Resource Kit  
Network Management Tools

Megkérdőjelezhetetlenül nagy hiányossága volt az eredeti terminálszolgáltatás kliensének a kiszolgáló és a helyi gép közötti állományműveletek megoldatlan kérdése. Mi a helyzet ha SMB kapcsolat nélkül csatlakozom a kiszolgálóhoz? Ha más állományátviteli (FTP, WEB-DAV, stb) szolgáltatás nincs engedélyezve? Erre ad választ ez a hiánypótló kiegészítés, amely remélhetőleg a következő szervizcsomag részeként mindenhova el fog jutni. Egyelőre be kell azonban értnünk ezzel a körülményesen telepíthető bővítéssel, és figyelembe kell vennünk, hogy a weben publikálható ActiveX TSAC-t nem támogatja, kizárólag a hagyományos kient. Azonban megfelelően telepítve megoldja ezt a problémát, és immáron gondatlanul lehet másolgatni a kient futtató gép és a kiszolgáló állományrendszere között.

A kiegészítés egyaránt támogatja a 16 és a 32 bites klienseket, a kivágás, másolás és beillesztés funkciókat mind egyedi állományok, mind teljes könyvtárstruktúrák esetében is. A hagyományos Ctrl+C Ctrl+V billentyűkombinációk tovább élnek, ugyanúgy, mint a Windows Explorer idevágó Copy, Cut, Paste utasításai. Ez a funkció a Windows 2000 terminálszolgáltatás virtuális csatorna architektúrájának köszönhetően működik. A telepítéséhez kient és a kiszolgáló oldalon egyaránt kell változtatnunk a terminál-szoftverben. A kiszolgáló oldalon vagy a már fellelpezített Resource Kit könyvtárból (egy terminálkapcsolaton keresztül) futtassuk le az fxfr.inl.bat parancsállományt, vagy manuálisan tegyük meg a következő lépéseket:

1. A WINNT\SYSTEM32 könyvtárban hozunk létre egy RDP CLIP alkönyvtárat.
2. Másoljuk ide a kicsomagolt Resource Kit könyvtárból az RDPCLIP.exe állományt.
3. Egészítsük ki a regisztrációs adatbázist az fxfr.inl tartalmával. Ezt manuálisan a regedit használatával, vagy a „regini fxfr.inl” parancsral tehetjük meg.
4. Másoljuk a WINNT\SYSTEM32 könyvtárhoz a kicsomagolt Resource Kit forrásból az fxfr.dll állományt.

Kliensoldalon a Resource Kit könyvtárból először az fxfr.dll állományt kell bemásolni a Terminal Server Client könyvtárba, ahol az mstsc.exe is található. Ez alapértelmezés szerint a C:\Program Files\Terminal Server Client elérési úton található. Ezután az rdpr.dll kerüljön szintén ugyanide, felülírva az eredeti rdpr.dll-t.

A MICROSOFT MAGYARORSZÁG SZAKMAI MAGAZINJA  
 2001. 04.

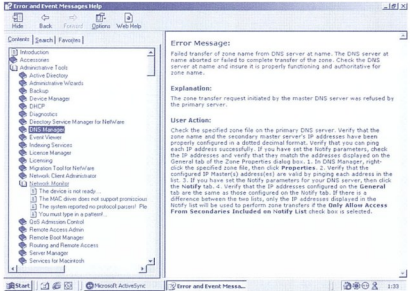


Végül már csak a kiszolgáló és a kliens újraindítására van szükség. A dokumentációban egy nem létező, `fxfr_srv` reg nevű állományra hivatkoznak, amely valójában az `fxfr.ini` formájában kerül fel a CD-re. *(Hogy én ezzel mit kintlódtam! Natív spekulációval nem jutottam eredményre. Hiába no, olvasott embernek párgja nincs :) – a szerk.)*

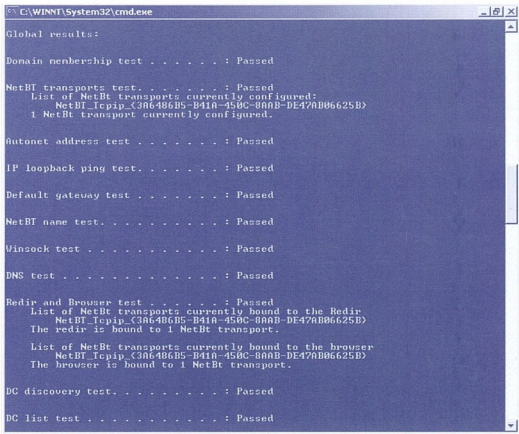
**Windows 2000 Error and Event Messages Help**

Windows 2000 Support Tools  
Computer Management Tools  
W2000msgs.chm

Egy átlagos, viszonylag problémamentes (*most tessék lekopogni*) hálózat üzemeltetése során is rendszeresen találunk hibáuzeneteket, a lehető legváratlanabb helyeken, időnként meglehetősen abszurd tartalommal. Ezek megfejtése erősen emberséges próbálódás a rendszerfelügyeleti munka. Azonban itt a varázslatos szótár, amely összefoglalja minden kategóriázva, kereshetően. Az adott bejegyzés a hiba leírásán kívül tartalmazza a megoldási javaslatot. Természetesen nem kizárólagos forrásként kezelendő, a különböző Technet és tudásbázis (*KB*) cikkek legalább annyira vonatkoznak lehetnek. És nemcsak az Eseménynapló bejegyzéseire hegyezték ki – a legtöbb gyári Windows alkalmazásra és parancsra vonatkozóan van bejegyzés. Legyen az Backup vagy Terminal Service, esetleg netstat vagy route, bejegyzések bőven állnak rendelkezésünkre. Még Marcell kedvenc Network Monitorjáról is találtam ezt-azt. Nem állítom hogy automatikusan mindenre választ kaphatunk itt, de segítségével legalább eggyel több esélyünk van a lelki békére.



általában rejtélyes, nehezen debuggolható ügyben küzdelmet kell vívniuk a rendszerrel. Van hogy egy eszközmeghajtó kavarja össze életünket, van hogy a kiszolgálók szolgáltatási oldalán nincs valami rendben. A program paraméterezés nélkül futtatta is max 5 másodpercig dolgozik, ekkor minden alapvető teszt lefut. Ezáltal könnyen kiokíthatjuk akár felhasználóinkat is a parancs egyszerű futtatására, amelyből gyorsan kiderül, hogy mi nem stimmel. A teljes tesztfelSORolás helyett most kiemelnék néhány fontosabb vizsgájt tényező: DNS, WINS kiszolgálók, Kerberos, NDIS, Winsoc, alapértelmezett átjáró. Ellenőrzi a gép tartomány-tagságát, valamint a WAN paramétereket is egyből kezeli. Az összes egyenértékű információ kinyeréséhez 5-10 különböző parancsot kéne egy-ébként kiadnunk (`ipconfig -all; ping dns.domain.hu; ping gateway.domain.hu; stb`), vagy külön hardvertesztelő eszközt bevetni (*lásd NDIS teszt*). Használatának alapfeltétele legalább egy hálózati eszköz, amely TCP/IP csatlósall rendelkezik. Emellett vizsgálja az IPX kapcsolatot is, valamint Netware kliens esetében az NDS vagy bindery bejelentkezést. Hihetetlen, de mindemellett még Microsoft multiplatformos is, Windows 95, 98 és NT 4.0 rendszerekben is működik a Windows 2000 mellett.



☞ **Hálózati kapcsolatunk szép és egészséges**

☞ **Help! Help!**

**Hálózati kapcsolat vizsgálata: Netdiag.exe**

Windows 2000 Support Tools  
Diagnostic Tools

A Windows 2000 igen megbízható hálózati implementációjának és beállításainak (*protokollok, kártyák és szolgáltatások*) működésével ritkán akad baj. Ha viszont problémák vannak,

Következő alkalommal a cím tárcsá felé ezvelek, az AD és a hozzá kapcsolódó területekről (*LDAP, ADSI*) választok néhány segédprogramot.

*Radványzski Gábor, MCP  
rgabor@sved.net*

**A cikkben szereplő URL-ek:**

- [1] <http://msdn.microsoft.com/subscriptions/resources/subdwnd.asp>
- [2] <http://www.microsoft.com/windows2000/library/resources/reskit/tools/default.asp>
- [3] <http://windows.microsoft.com/windows2000/en/server/iis/htm/core/iicodira.htm>

# SMTP az Exchange 2000-ben



Az Interneten zajló levelezés alapja a SMTP Protokoll (*Simple Mail Transfer Protocol*). A Microsoft Exchange Server mindig is képes volt üzeneteket küldeni más SMTP kiszolgálóknak: az Internet Mail Connector (*IMC*) egy opcionális kiegészítő komponens az Exchange Server 4.0 rendszerhez, az Internet Mail Service (*IMS*) pedig valamennyi Exchange Server 5.0 és 5.5 kiszolgálóban megtalálható kiegészítő. Bár könnyen telepíthető az IMS, az SMTP az Exchange Server korábbi verzióiban nem játszott kulcsszerepet. Az IMS hasonlít a többi csatlóhoz, amelyekkel más levelezőrendszerhez (pl. Lotus Notes, ccMail, IBM PROFS) tudjuk csatlakoztatni az Exchange kiszolgálónkat, és szorosan együttműködik az X.400-alapú Message Transfer Agenttel (*MTA*), mely az üzenetek vállalaton belüli irányításáért felelős. Az útválasztófeladatokat az Exchange Server korábbi verzióiban ezt a feladatot az MTA látta el. Az Exchange 2000 Server rendszerben teljesen új, SMTP alapú útválasztás jelent meg.

Az az architektúra, mely a különböző csatlókat az MTA köré fogta össze az Exchange Server kezdeti verzióiban akkor jelent meg, amikor az X.400 volt az üzenetküldés ipari szabványa. Mostanra az SMTP foglalta el a helyét – pontosabban annak továbbfejlesztett, napjaink legjelentősebb üzenetkezelő rendszerei által alkalmazott verziója. Az SMTP az Exchange 2000 üzenetovábbító rendszernek alapja, mely az Exchange Server korábbi verzióiról történő átállás során is biztosítja a levelezés folyamatos működését.

Az Exchange 2000 ismertetése előtt fontos megvizsgálni az SMTP történetét és fejlődését, integrálódását az Exchange Server rendszerrel, útválasztó és adminisztrációs szabályait, valamint jövőjét az üzenetovábbítás területén. Ismerni kell továbbá, hogy az útválasztócsoporthoz miként fogják logikai csoportba az egyes kiszolgálókat az üzenetküldés folyamatához.

## Az SMTP története és fejlődése

Az IETF 1982-ben definiálta az SMTP jellemzőit a 821-es és 822-es ajánlásokban (*RFC*). Nevének megfelelően az SMTP eredeti verziója egyszerű volt, a fő cél síma 7-bites karakterkészletű szöveges üzenetek küldése egy kiszolgáló és egy ügyfél között. Az SMTP műveletek alapértelmezett portja a 25-ös lett. 1982 óta az SMTP folyamatosan fejlődött, hogy napjaink modern üzenetovábbítási követelményeinek megfeleljen. Értethető okokból a fejlődés az Internet terjedésének hatására az elmúlt években különösen felgyorsult. A két legfontosabb újítás, melyek segítségével az SMTP a legmodernebb igényeket is képes kielégíteni a továbbfejlesztett SMTP (*Extended SMTP, ESMTP*) és a MIME (*Multipurpose Internet Mail Extension*), melyről a lapszámunk 33. oldalán bővebben is olvashat.

## ESMTP

Az ESMTP lehetővé teszi, hogy a gyártók az alap SMTP szolgáltatást kiegészítővel lássák el, mely így képessé válik különböző új üzenetfajták és funkciók kezelésére. Az Ex-

change Server 5.5 is támogatja az ESMTP-t, mivel lehetővé teszi az Internetszolgáltató levelezőrendszerén lévő üzenetek egyszerűbb átvételét. Erre akkor van szükség, ha valaki az Internethez egy szolgáltatón keresztül csatlakozik, és nem rendelkezik folyamatos kapcsolattal. A szolgáltató tárolja a leveleket, melyeket később az Exchange segítségével le lehet tölteni, és szét lehet osztani a helyi felhasználók között. Az Exchange Server korábbi verziói támogatnak néhány ESMTP parancsot. Például az SMTP telefonos hálózaton történő használatához az Exchange Server 5.5 támogatja az ETRN parancsot, mely az üzenetsorok kezelését is egyszerűsíti. Az Exchange Server és más SMTP kiszolgálók közötti hitelesített és titkosított kapcsolat támogatásához az Exchange Server 5.5 Service Pack 2 (*SP2*) és későbbi verziói ismerik az AUTH=LOGIN, STARTTLS, és TLS kiegészítéseket. Egy SMTP kiszolgáló által nyújtott ETRN támogatás szintje könnyen meghatározható, ha telnet programmal a kiszolgáló 25-ös portjára csatlakozunk, majd kiadjuk az EHLO parancsot. A kiszolgáló ekkor megjeleníti az általa támogatott kiterjesztések kulcsszavait. Az első kép az Exchange Server 5.5 SP3 által nyújtott, a második az Exchange 2000 által biztosított kiterjesztéseket mutatja. A legfontosabb új kiegészítések a CHUNKING, PIPELINING, DSN, és a X-LINK2STATE (melyről a későbbiekben még szó lesz).

```
Command Prompt: telnet
250 exchange.sqsoft.hu ESMTP Server (Microsoft Exchange Internet Mail Service)
250 250-215 ready
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
```

## Az Exchange Server 5.5 SP3 SMTP tudása

```
Command Prompt: telnet
250 bangalore.asia.ntrvaders.msf Microsoft ESMTP MAIL Service, Version: 5.0.211
250 250 ready at Mon: 2 Apr 2001 08:31:38 +0700
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
```

## Az Exchange 2000 SMTP tudása

A Chunking támogatja a stream-módu adatátvitelt, vagyis az Exchange 2000 nemcsak soronként képes üzeneteket küldeni és fogadni, hanem bináris adatként is. A korábbi SMTP kiszolgálók, mint az IMS, az SMTP üzeneteket sorokból álló szöveggként kezelték, nem összetett struktúráként,

mint azt ma látjuk. Az üzenetek bináris alapon történő kezelése teljesítményszempontok miatt fontos - különösen az egyre növekvő csatolt fájloknál.

- ☞ A Pipelining is a teljesítményt növeli. Ennek révén az SMTP kiszolgáló több parancsot adhat ki egyszerre, nem kell várakoznia minden egyes parancs visszaigazolására.
- ☞ A Kézbiztosítási állapot értesítés (*Delivery Status Notification, DSN*) lehetővé teszi, hogy az Exchange 2000 szabványos értesítést küldjön a felhasználóknak a kézbesítési sikerességéről. Ezek az értesítések veszik át az Exchange Server 5.5-ben megjelenő nem szabványos hibajelzéseket, melyeket egy üzenet sikertelen kézbesítése okozott.

## MIME

Az egyszerű szöveges üzenetek unalmassá válhatnak, főként napjaink grafikus világában. A korai kódolási eljárások, mint a UUencode lehetővé tették csatolt fájl és formázott szövegek küldését, de ezek a lehetőségek rendszerint nem voltak egységesen integrálva. Továbbá a különböző csatolt fájlok által használt formázó jeleket pontosan be kellett állítani, hogy a fogadó fél értelmezni tudja azokat.

A MIME megoldja ezt a problémát, mivel szabványos címkek segítségével szabályozza az üzenetekben lévő különböző típusú adatok jelölését és továbbítását. Alapvetően a MIME azt mondja meg a levelezőrendszernek, hogy az miként dolgozza fel az üzenet egyes részeit, hogy a fogadó pontosan azt kapja meg, amit a küldő neki szánt. Talán használta már a MIME-t Microsoft Word dokumentumok vagy Microsoft PowerPoint prezentációk küldésére. A MIME az alapja az audio-és videoanyagok üzenetként történő küldésének is, melyek általában sokkal nagyobbak, mint a dokumentumok vagy prezentációk. Híres példa a Baljós Árnyak (*Phantom Menace*) 25 Gigabájtos előzetesének esete, mely sok levelezőrendszert terhelte le a végsőig, amikor a felhasználók elkezdték azt egymásnak küldözgetni. A csatolókon alkalmazott korlátozások megakadályozzák nagyméretű csatolt állományok küldését más cégek felhasználóinak, ugyanakkor a nagy üzenetek a belső levelezésben is gondokat, késedelmet okozhatnak. A rendszergazdák jogosan tartanak attól a terheléstől, amit a nagy üzenetek jelentenek a kiszolgálóknak, ezért próbálják lebeszélni a felhasználókat a nagy hang- és mozgókép üzenetek küldéséről. Ennek ellenére a nagy csatolt fájlok a jövőben egyre gyakoribbak lesznek, számítani kell arra, hogy hálózaton egyre több nagyméretű csatolt fájl lesz jelen.

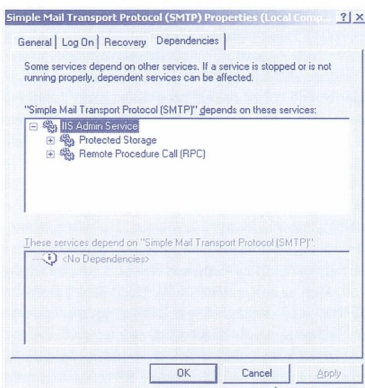
## Az SMTP és az Exchange 2000

A Microsoft az Exchange 2000 fejlesztésénél minden, kiszolgálók közti kommunikációt SMTP alapon szervezett meg – ez radikális változtatás az addig használt távoli eljáráshiváshoz (*remote procedure calls, RPCs*) képest. Az RPC-vel történt szakítás helyes döntés volt. Amikor működik, a távoli eljáráshivás remek eszköz, de nagy sávszélességet igényel a folyamatos használatra. Amikor nem működik, hatalmas üzenetsorok gyűlnek fel nagyon gyorsan. Az RPC-k mellőzése része az Exchange 2000 fejlesztése során tapasztalható általános tendenciának, hogy a Microsoft-specifikus protokollokat szélesebb körben használt Internetes protokollok cserélik le. Az Exchange 2000 csak akkor használ RPC-t, amikor vegyesmódú hálózatban másik kiszolgálóhoz csatlakozik. (*Vegyesmódú hálózatban mind Exchange 2000,*

*mind korábbi verziók megtalálhatók.*) Ez logikus, mivel az RPC jelenti az egyetlen protokollt, melyet minden Exchange kiszolgáló garantáltan ismer.

Az Exchange Server korábbi verzióiban az SMTP támogatást külön kellett telepíteni és beállítani. Az Exchange 2000 kiszolgálóknál az SMTP már automatikusan telepítésre kerül. Alapvetően minden Windows 2000 (*Win2K*) kiszolgáló telepíti az SMTP szolgáltatás egy egyszerű fájtaját, így az alkalmazások adatokat cserélhetnek a kiszolgálók közt. A Win2K ezzel is végezheti a kiszolgálók közti Active Directory replikációt. Az Exchange 2000 telepítése során az alap SMTP szolgáltatás számos új elemmel bővül, mint az Exchange Information Storeba (*levelezésközpontba*) történő kézbesítés, vagy a fejlett üzenet-sor-kezelési mechanizmus. Technikailag az Exchange 2000 frissíti az SMTP szolgáltatást, mely így a Collaboration Data Objects (*CDO*) 3.0-s verzióját is támogatni képes a 2.0 verzió után. A fejlett üzenet-sor-kezelési mechanizmus korlátozások bevezetését teszi lehetővé (*mind a levelezésközpont, mind a csatolókra*). Az első Exchange 2000 kiszolgáló telepítése frissíti az Active Directory sémáját, hogy – többek között – az új megszorításokhoz szükséges attribútumokat támogassa. Az Exchange Server 5.5 által támogatott levelezésközpont kivül mostantól meghatározható például, hogy a nagy méretű üzenetek mely időpontokban haladhatnak át egy csatlókon.

Az operációs rendszer és a levelezőrendszer szoros integrációját mutatja az a tény is, hogy az Exchange 2000 csak kiterjeszti, de nem újratelepíti a Win2K meglévő, alapított SMTP szolgáltatását. Bár az Exchange Server 5.5 és a Windows NT közt is szoros a kapcsolat, az Exchange 2000 és a Win2K számos más ponton is kapcsolódik egymáshoz. Ez a sok összekapcsolódás meg is nehezítheti a rendszer felügyeletét, és a folyamatok részletes ismeretét teszi szükségessé. Az Exchange 2000 rendszer sikeres üzemeltetéséhez tisztában kell lenni valamennyi függőséggel és kapcsolattal.



## ☞ SMTP Tulajdonságok

A fenti ábra mutatja, hogy az SMTP szolgáltatáshoz szükség van a Microsoft Internet Information Services (*IIS*) szolgáltatásra is. Az IIS problémái befolyásolhatják az SMTP szolgáltatást is. Hasonlóképpen, ha az SMTP szolgáltatással gond van, elképzelhető, hogy nem lesz képes üzeneteket

küldeni más Exchange kiszolgálóknak, vagy a kívüllagnak. Az SMTP hibás működése más alkalmazásokat is érinthet. Például a címárak közti replikáció leállhat, ha egy tartományvezérlő SMTP beállításai hibásak, mivel a Win2K telephelyek közti replikáció SMTP üzenetek segítségével is történhet.

### Virtuális kiszolgálók

A Win2K-ban az internetes protokollak - HTTP, az SMTP, a POP3, és az Internet Message Access Protocol 4, (*IMAP4*) - az IIS 5.0-be vannak integrálva, és így mindegyik protokollal létre lehet hozni virtuális kiszolgálót. Egy virtuális kiszolgálót felfoghatunk úgy, mint a protokoll, a port és a konfigurációs adatok összessége. Egy gépen több virtuális kiszolgáló futtatható, vagyis egy Exchange 2000 kiszolgálón akár több virtuális SMTP szerver is működhet egyszerre. Az alaphelyzet szerinti virtuális kiszolgáló a 25-ös portot használja, de a többi virtuális kiszolgáló akár más porton is működhet, illetve egyéb más beállítások adhatók nekik. Ez a funkció lehetővé teszi, hogy egy Exchange 2000 kiszolgáló számos különböző típusú és igényű e-mail tartomány levelezését lássa el, és erre az Internetszolgáltatóknak nagy szükségük is van. Továbbá az Exchange 2000 támogatja több SMTP csatló (*SMTP Connector*) kialakítását egyetlen virtuális kiszolgálón. Ha az IMS-t is egy SMTP csatlóknak tekintjük, akkor az Exchange 5.5 esetében egy gépen csak egyetlen SMTP csatló lehet, de az Exchange 2000 segítségével több csatló állítható be, mindegyik különböző célra. Például az egy csatló a levelek belső hálózatra történő irányításáért felel, egy másik pedig a kimenő leveleket egy Smart Host felé irányítja, ahol a levelek virusellenőrzésen esnek át, mielőtt az Internetre kerülnének.

### Az üzenetkezelés adminisztrációja

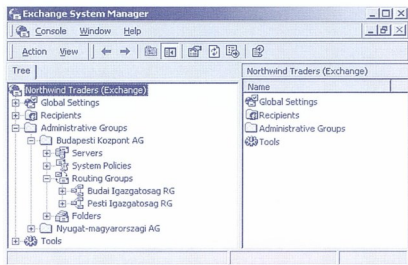
Az Exchange Server 5.5 a kiszolgálókat telephelyekre (*site*) fogja össze - mely egy vagy több, helyi hálózatnak megfelelő sávszélességen csatlakozó kiszolgálót tartalmaz. A telephelyek összessége alkotja a szervezetet (*organization*), egy felületi egységet alkotva, mely megosztott címárakkal és konfigurációval rendelkezik. A szervezeti koncepció megmaradt az Exchange 2000 rendszernél is. Egy Win2K hálózatban csak egyetlen Exchange 2000 szervezet létezhethet, mivel az Active Directoryban a közösen használt konfigurációs partícióban az Exchange szervezetek konfigurációs adatai részére egy tároló hely van fenntartva.

A Win2K terminológiája a telephely (*site*) kifejezéssel a közös hálózaton osztozó kiszolgálókat jelöli - ebben az esetben egy vagy több IP alhálózat alkotja a hálózatot. A Win2K telephelyek sokkal szorosabban kapcsolódnak az alacsonyabb hálózati rétegekhez, mint az az Exchange Server 5.5 telephelyek esetén volt. Kicsi az esélye annak, hogy olyan Win2K telephelyet találjunk, mely az egész világot átfojja, ugyanakkor számos ilyen Exchange Server telephelyről tudunk. A Win2K telephelyek kialakításának nincs olyan közvetlen adminisztratív következménye, mint az Exchange Server 5.5 telephelyeknek, amelyek egyben az adminisztrációs, biztonsági és replikációs műveletek határait is jelentik.

Az Exchange 2000-ben az adminisztrációs és útválasztó csoportok (*administrative group, routing group*) váltották fel az Exchange Server 5.5 telephelyeit. Az adminisztrációs csoport jelenti a felületi műveletek kereteit. A rendszergazda egy adminisztrációs csoportba tartozó valamennyi kiszolgáló felügyeletéért felelős. Az útválasztó csoportok határozzák

meg, hogy az üzenetek milyen útvonalon közlekedjenek a szervezet kiszolgálói közt. Egy útválasztó csoporton belül bármely kiszolgáló igény szerint pont-pont kapcsolat alakíthat ki egy másik kiszolgálóval, pontosan úgy, ahogy az Exchange 5.5 telephelyben a kiszolgálók csatlakoznak egymáshoz. Vegyes módú Exchange szervezetben az adminisztrációs csoportokon belüli útválasztó csoportok felelnek meg az Exchange 5.5 telephelyeknek, azaz az Exchange 2000 úgy látja a saját csoportosításait az Exchange 5.5 kiszolgálók számára, mintha a telephely-modellt használná, mivel a korábbi kiszolgálók nem ismerik az adminisztrációs és az útválasztó csoportot. Később, amikor minden kiszolgálón az Exchange 2000 fut, a szervezet natív módba kapcsolható, és az útválasztó csoportokból új struktúra alakítható ki, például olyan, hogy egy útválasztó csoport más adminisztratív csoport kiszolgálóit is tartalmazza. A telephelyek nagyon hatékonyak kis szervezeteknél, ahol a rendszergazda feladata tipikusan a felügyelet és az irányítás figyelése. Nagyobb szervezeteknél azonban gyakran külön csoportok felelősek a kiszolgálók felügyeletéért és a levelezés biztosításáért. Jó hír, hogy amennyiben a vállalat működése azt sugallja, akkor az Exchange 2000 pontosan úgy felügyelhető, mint az Exchange Server 5.5. Az Exchange 2000 alkalmazásában azonban lehetőség van arra is, hogy a felelősséget különböző csoportok közt osszuk szét. Az adminisztrációs feladatok hatékonyabb szétosztása kiemelt fontosságú a Win2K rendszerekben. Ezért készítette el a Microsoft a Microsoft Management Console (*MMC*) keretrendszer, melynek segítségével testre szabott konzolok készíthetők a szükséges komponensekből. Sok nagy társaság több szinten üzemelteti a számítógépes rendszert. Az Exchange Server 5.5 egyetlen, mindenki által használható egyszerű eszközt biztosít - a Microsoft Exchange Administrator programot. Általában kényelmes megoldás ugyanazt az eszközt használni minden feladatra, de miért kellene ágyút adni annak a kezébe, aki csak egy verébre lő (*egy felhasználó adatait szeretné megváltoztatni, hozzá kell férnie egy összetett komponens beállításához stb.*)?

Az alábbi ábra az Exchange 2000 System Manager mutatja, egy MMC konzolt, mely számos bővítményt tartalmaz. Ez az eszköz hasonlítható leginkább az Exchange Administratorhoz. Valójában az egyes bővítmények akár különböző üzemeltetői csoportokhoz rendelhetők. Az ábrán az adminisztrációs és útválasztó csoportok közti kapcsolat látható. Egy adminisztrációs csoport tipikusan nagyobb területet ölel fel, mint egy útválasztó csoport. Ebben a példában két adminisztrációs csoport felügyel egy hálózatot, és mindegyik rendelkezik egy vagy több útválasztó csoporttal.



► Exchange 2000 System Manager

## Útválasztó csoportok

Minden kiszolgáló, ami egy útválasztó csoportba tartozik automatikusan képes egymással kommunikálni. Ez így volt az Exchange Server 5.5 telephelyeknél is, de itt a kiszolgálók más-ként kommunikálnak. Emlékeztetül, az Exchange Server 5.5 kiszolgálók távoli eljárásíhvást alkalmaznak, míg az Exchange 2000 esetében a kiszolgálók SMTP használatával, alapesetben a 25-ös porton kommunikálnak. A másik alapvető különbség az Exchange Server 5.5 és az Exchange 2000 közt, hogy Exchange Server 5.5 esetén a kiszolgálók közti címárreplikáció teszi ki a forgalom jelentős részét, míg Exchange 2000 kiszolgálóknál a forgalmat csak a felhasználók személyes üzeneti és a nyilvános mappák replikációs üzenetei alkotják, mivel az Exchange 2000-nec nincs önálló címár, az Active Directory-t használják. Az Exchange 2000 kiszolgálók csak a hálózat állapotát leíró kis adatcsomagot küldenek egymásnak, de ez jóval kisebb forgalmat jelent, mint a telephelyek címárreplikációja. A címár frissítése most az Active Directory feladata.

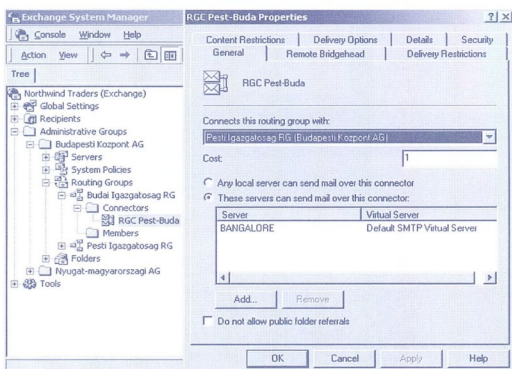
## Az útválasztó csoportok csatlakoztatása

Útválasztó csoportok csatlakoztatása az SMTP csatlakoztatás, az X.400 csatlakoztatás, vagy Routing Group Connector (RGC) alkalmazásával történhet. Bizonyos szempontból az RGC megfeleltethető az Exchange Server 5.5-ben alkalmazott távoli eljárásíhvást alapú Site Connectornak. Az RGC-t egyszerű telepíteni és beállítani. A Site Connectorral ellentétben az RGC szélesebb körű beállítási lehetőségeket biztosít a levéltovábbításban résztvevő kiszolgálók felett. A Site Connector segítségével egy telephely valamennyi kiszolgálójának vagy egy megnevezett hídfőkiszolgálójának (bridgehead server) a forgalmát lehet szabályozni. Az RGC lehetővé teszi, hogy bármennyi kiszolgáló hídfőként működjön. Az RGC továbbá beállítható úgy, hogy a csatlakoztat csak bizonyos felhasználók használhassák, a csatlakoztatásmenő forgalom prioritású is hangolható (pl. magas, közepes, alacsony). Ezen kívül lehetőség van időzítések megadására, hogy a nagy üzenetek mikor haladhasanak át a csatlakoztat, így a nagy csatlakoztat állományok nem akadályozzák a normális üzleti forgalmat.

Az RGC az SMTP forgalmat kezeli, és egy virtuális SMTP kiszolgáló felelt fut. Az RGC egyirányú csatlakoztat, az üzenetek mindkét irányba történő közvetítéséhez először külön be kell állítani mindkét oldalt. Ha a megcélzott útválasztó csoportban adminisztrátori jogosultsággal rendelkezünk, a csatlakoztat mindkét oldala egyetlen lépéssel beállítható.

Az alábbi ábra mutatja a Budai Igazgatóság és a Pesti igazgatóság útválasztó csoportot összekötő RGC kapcsolat tulajdonságait. A Do not allow public folder referrals (Nyilvános mappára történő hivatkozás tiltása) jelölőnégyzet igen fontos. Alapesetben ez nincs kiválasztva, tehát a felhasználók ezen a csatlakoztat keresztül másik útválasztó csoportban lévő nyilvános mappákhoz is hozzáférhetnek. Az Exchange 2000-ben a „hivatkozások” lépnek az Exchange Server 5.5 alatt alkalmazott nyilvános mappá „affinitás” koncepció helyére, és most az útválasztó csoportok vették át a telephelyek szerepét. Amikor a felhasználó egy nyilvános mappához kíván hozzáférni, az Exchange 2000 először a felhasználó postaládáját tároló kiszolgálón beállított nyilvános mappá kiszolgálón próbálja elérni a nyilvános mappá egy példányát. Ha ott nem talál ilyet, az Exchange 2000 a felhasználó útválasztó csoportjával azonos csoportban lévő kiszolgálóknál próbálko-

zik. Ha ott sem talál másolatot, az Exchange 2000 a legalacsonyabb költségű csatlakoztat keresztül csatlakoztat egy másik útválasztó csoporthoz, és ott próbálkozik. Minden csatlakoztat rendelkezik egy költségétényezővel (1-től 100-ig) ami a kapcsolat (hálozati értelemben vett) költségét jelzi egy hídfőhöz való csatlakoztatásakor. Az alacsonyabb költségű útvonalak előnyben élveznek. Az alábbi ábrán látható RGC-nél a csatlakoztat költsége 1, ami a lehető legalacsonyabb érték. Ez a csatlakoztat egy preferált irány egy nyilvános mappá másolatához, ha az a Pesti Igazgatóság útválasztó csoportban megtalálható.

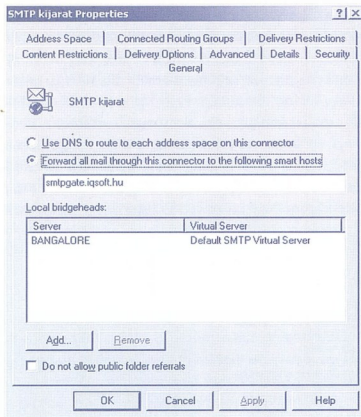


## ◀ RGC tulajdonságok

Ha az RGC SMTP-t használ, miért van szükség külön SMTP csatlakoztat? A válasz egyszerű – az Exchange 2000 rendszerek egy kicsi részét képezik az SMTP-t használó kiszolgálóknak. Az emberek más SMTP-t használó rendszerekkel is kommunikálni akarnak, és ezt olyan egyszerűen szeretnék tenni, ahogyan csak lehet.

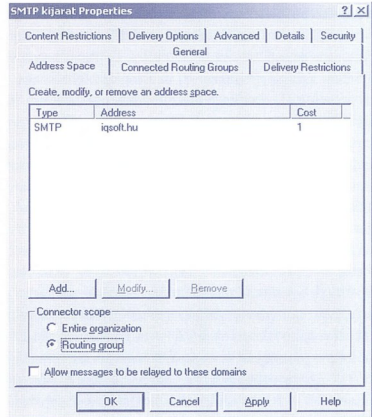
Az RGC-hez hasonlóan az SMTP csatlakoztat is egy virtuális SMTP kiszolgáló felelt fut. Míg az RGC az Active Directory adatait használja az üzenetek irányításához, addig az SMTP csatlakoztat olyan szabványos SMTP útválasztó technikákat alkalmaz, mint a névkiszolgálók (DNS) levéltovábbítás (mail exchanger, MX) rekordjai. Ezért, ha a leveleket Smart Host rendszerhez szeretnék irányítani, vagy MX rekord alapú irányítást kívánunk alkalmazni, vagy csak egyszerűen üzeneteket kívánunk küldeni az Exchange kiszolgálón túlra (beleértve a korábbi Exchange kiszolgálón futó IMS rendszereket is), akkor szükségünk van egy SMTP csatlakoztat.

Az alábbi ábra egy SMTP csatlakoztat mutat. Ez a csatlakoztat valamennyi üzenetet az smtpgate.iqsoft.hu címre irányítja. A Smart host olyan levelező-kiszolgáló, mely legtöbbször az adott vállalat SMTP levelezésének központja. Az üzenetek kézbesítés előtt egy helyre történő összegyűjtése lehetőséget biztosít pl. vírusellenőrzésre. A Smart host továbbá lehetővé teszi annak ellenőrzését, hogy a felhasználók betartják a vállalati szabályokat (pl. általános információk fűzése a levélhez, bizalmas adatok küldésének megakadályozása, stb). A MIMesweeper (<http://www.mimesweeper.com>) egy népszerű példája a smart hostokon futó szoftvereknek.



#### ◀ SMTP csatoló

Valamennyi üzenet Smart host-hoz történő irányítása csak az egyik beállítási lehetőség. További beállítások adhatók meg az SMTP csatoló címter (Address Space) tulajdonságlapján, ahol megadhatjuk, hogy az Exchange 2000 mely SMTP tartományokat érje el az adott csatolón keresztül. Alapértelmezésben ez csillag (\*), vagyis az Exchange 2000 minden tartományt elérhet. Néhány esetben hasznos lehet, ha az Exchange 2000 egyes tartományok üzeneteit egyedí módon kezeli. A következő ábra mutatja egy adott tartományhoz (pl. iqsoft.hu) rendelt címter beállításokat. Ezzel a beállítással a csatoló csak az iqsoft.hu-hoz tartozó címeket kezeli, a többi üzenetnek más útvonalat kell találnia. Az ábra alján látható, hogy az Exchange 2000 miként kezeli a csatoló hatáskörét. Az Exchange Server 5.5 vezette be a csatoló hatásköre (Connector scope) fogalmat, ami egy csatoló elérhetőségét jelöli más kiszolgálók számára. Az Exchange Server 5.5-ben a csatolók elérési terület, tartomány vagy szervezeti alapon lehetett korlátozni. Az Exchange 2000 esetében a megszorítás útválasztó csoportra vagy az egész szervezetre állítható be. Az ábrán a csatoló csak az azonos útválasztó csoportba tartozó kiszolgálók számára érhető el.



#### ◀ SMTP Címter

#### Kapcsolatállapot alapú útválasztás

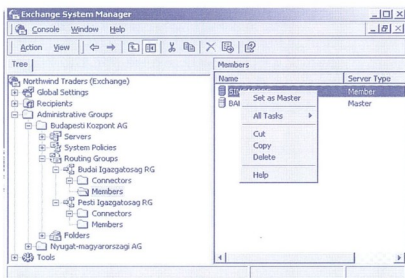
A csatoló egyszerűen egy kapcsolat egy útválasztó csoport és más útválasztó csoport vagy más típusú rendszerek közt. Egy vállalaton belül több útválasztó csoport és csatoló működhet. Azért, hogy az üzenetek irányítása hatékony legyen, egy olyan mechanizmusra van szükség, ami a kiszolgálókat informálja a többi kiszolgáló és csatoló pillanatnyi állapotáról. Egy szervezeten belüli útvonalak nyilvántartásához az Exchange Server 5.5 a Gateway Address Routing Table-t (GWART) alkalmazza. A Relative Identifier (RID) kiszolgáló (tipikusan ez az elsőként telepített kiszolgáló) felépíti a GWART-ot a szervezetben replikálódó címteradatok alapján. A telephely többi kiszolgálója közösen használja a GWART-ot, így minden kiszolgáló rendelkezni fog az üzenetek hatékony irányításához szükséges információkkal.

Az Exchange Server 1996 óta ezt a modellt követte. A GWART nagyon sikeres, de statikus jellege komoly hátrányt jelent. A GWART akkor működik hatékonyan, ha a hálózat stabil, a kiszolgálók és csatolók kiszámítható módon viselkednek. Ha nem kiszámítható faktorok jelennek meg az irányításban, akkor a GWART hatékonysága csökken, és a téves irányítások száma növekedni kezd. A téves irányítás nem jelenti az üzenet elvesztését, azonban az üzenet zsákutcába kerülhet. Ekkor újra kell irányítani, hogy végső céljába jusson. Az újirányítás drága és lassítja az üzenet kézbesítését, tehát kerülendő.

Mivel a stabil hálózati környezet nem mindig megvalósítható, dinamikusabb módszerre van szükség a kiszolgálók értesítéséhez a hálózat aktuális állapotáról és az elérhető útvonalakról. Az Exchange 2000 kapcsolatállapot alapú útválasztást (Link State Routing, LSR) alkalmaz. Az LSR segítségével a kiszolgálók értesítik egymást a hálózati kapcsolatokban és csatolókban bekövetkező változásokról. Ezek az információk gyorsan eljutnak más útválasztó csoportokhoz is, így az egész szervezet összes kiszolgálója mindig pontos képpel rendelkezik a hálózat állapotáról. Tegyük fel, hogy egy hálózati hiba miatt egy útválasztó csoporthoz való csatlakozás meghiúsul. Egy perc múlva a hídfő-kiszolgáló újból megpróbál a távoli útválasztó csoport hídfő-



kiszolgálójához csatlakozni, hogy kiderítse, vajon a hiba átmeneti volt-e. Ha a csatlakozás nem sikerül, a hídfőkiszolgáló még kétszer próbálkozik, mielőtt rögzítené hogy a probléma tartós. A kiszolgáló végül „belátja”, hogy a vonal nem működik, és a 691-es porton keresztül csatlakozik saját útválasztó csoportjának főnökéhez (*Routing Group Master, RGM*). Minden csoportban van egy útválasztó csoport vezérlő, amely az útválasztó csoportba tartozó kiszolgálók információit gyűjti össze, és tartja a kapcsolatot a távoli útválasztó csoportokkal. Ezen adatok alapján az RGM felépíti a Link State Table-t (LST), mely a hálózat aktuális állapotát tartalmazza. Az RGM típikusan az útválasztó csoport elsőként telepített gépe, de az Exchange System Manager segítségével ez a szerep könnyen átruházható, mit azt az alábbi ábra is mutatja.



#### ☞ Ki legyen a Routing Group Master?

Miután az RGM értesült a meghibásodott kapcsolatról, módosítja az LST-t, és tájékoztatja a változásról az útválasztó csoport többi kiszolgálóját. Még egyszer: a kommunikáció a 691-es port-on keresztül, speciális LSA protokollal történik, melyet a Microsoft direkt erre a célra fejlesztett ki. Az Exchange 2000 egy X-LINK2STATE nevű ESMTP parancsot használ a kapcsolatállapotok információinak frissítésére az útválasztó csoportok közt RGC és SMTP kapcsolónál egyaránt. Az Exchange 2000 a frissítéseket a hídfőkiszolgálóknak küldi, melyek azokat továbbítják az útválasztó csoport vezérlőknek (RGM). Az Exchange 2000 a frissítéseket alkalmazáspecifikus formátumban is tudja küldeni X.400 csatlakozáson keresztül. A továbbított adat nagyon tömörített,

így az Exchange 2000-nek csak egy pár másodpercre van szüksége, hogy továbbítsa az adatokat és az útválasztó csoport számára elkészítse az új LST-t. Új kiszolgálók vagy kapcsolatok telepítése szintén kapcsolatállapot frissítést eredményez.

A kapcsolatállapot frissítések gyorsan végbemennek az egész szervezetben. A cél az, hogy bármely kiszolgálót pár perc alatt értesíteni lehessen – a mai gyakorlattal ellentétben, ahol ez órákat, sőt napokat vehet igénybe. A kiszolgálók közt a frissítések olyan haladnak, mint az üzenetek. Mivel az Exchange 2000 csak kis mennyiségű adatot továbbít, a frissítések még azelőtt továbbításra kerülhetnek, hogy a rendszergazdák érzékelnék a problémát.

#### Az üzenetkezelés jövője

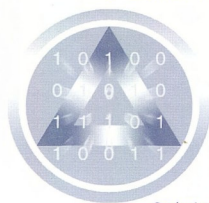
Itt csak néhány dolgot mutattam be az Exchange 2000 új technológiáiból. Az adminisztráció és az irányítás szétválasztásán túl az üzenetkezelő rendszer számos további lehetőséget kínál az SMTP szolgáltatással, és a virtuális kiszolgálókkal kapcsolatban. És ha ez nem lenne elég, a kapcsolatállapot alapú irányítás önmaga megérdemelne egy külön cikket.

Összefoglalva, az Exchange 2000 jelentősen különbözik elődeitől. Mivel az Exchange 2000 csak Win2K rendszereken fut, és az első Exchange 2000 kiszolgáló telepítése előtt típikusan már komoly infrastruktúra van jelen, az SMTP és az Exchange 2000 pontos megismerése komoly időt vehet igénybe. Azonban meg kell érteni, hogy az SMTP jelenti az egyik legszorosabb kapcsolatot a Win2K és az Exchange 2000 közt. Az üzenetirányítás jövőjét az Exchange kiszolgálókban az SMTP jelenti.

**Soós Tibor, MCSE, MCT**  
soost@iqsoft.hu

Szerzőnk az Exchange Server 4.0 megjelenésétől szakértője és oktatója az Exchange rendszereknek, az Exchange 2000 levelezőlista tanácsadója, az IQSOFT – John Bryce Oktatóközpont munkatársa.





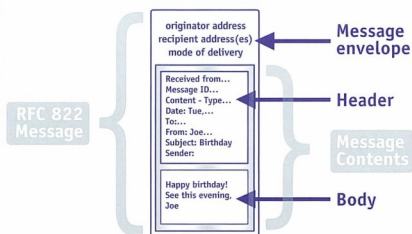
# Mime levelezés az interneten (I. rész)

Szabványrovatunk most a megszokottól eltérően alakul: a téma szerteágazó mivolta (és a rengeteg forrásanyag) miatt lapunk oldalain megpróbáljuk összefoglalni az internetes levelezés alapjait képező szabványokat és ajánlásokat, a nyolcvanas évek elején megalkotott, sptárain egyszerű RFC 822-től („Standard for the Format of ARPA Internet Text Messages”) egészen a napjainkban divatos, színes, szagos, néha zenélő HTML levelekig. Hogy kezdődött? Hova jutottunk? Mi a baj az ékezetekkel, a kódtáblákkal, a távol-keleti nyelvjárásokkal és a HTML levelekkel?

## RFC 822 – Szöveges üzenetek az Interneten

Az 1982-ben megjelent RFC 822 [1] meghatározta az Interneten (pardon, akkoriban talán még ARPANet-en) keresztül történő szöveges levelezés alapjait. Ez nem más, mint egy szöveges adathalmaz formátumának meghatározása, ami olyannyira sikerült, hogy a mai napig erre épül az Interneten küldött legtöbb levél. A levelek továbbítására használt SMTP protokoll leírását az RFC 821-ben [2] találja meg az olvasó, ennek tartalmára jelen cikk keretein belül nem térünk ki, de előfordulhat, hogy a cikksorozat elemei között találkozunk majd vele is. A protokoll megemlítése itt azért fontos, mert a levélformátumhoz hasonlóan az SMTP kommunikációs szabvány is fenntarad, és ez okozza a mai napi levelezésünkben felmerülő problémák nagy részét. De ne tessük az SMTP-t, egy közel húszéves protokoll nem tehet arról, hogy a világ levelezéskiszolgálóinak nagy részéből hiányoznak az azóta megjelent fejlesztések...

Visszatérve az RFC 822 üzenetre, az a következőkből áll össze:



### ☞ Egy SMTP üzenet felépítése

A teljes üzenet tehát egy „borítékban” (envelope) foglal helyet. Ezzel a borítékkal az átlag felhasználó általában nem találkozik, erre ugyanis a levelező kiszolgálók közötti kommunikáció során van szükség. A boríték tartalmát a levelet küldő kiszolgáló (vagy levelezőprogram) generálja, általában az üzenetbeli információk alapján: értesíti a célkiszolgálót, hogy ki a következő üzenet feladója (originator), címzettje (recipient), mi legyen a kézbesítés módja, satöbbi. Ez a kommunikáció voltaképpen az SMTP protokoll része. A dolog érdekes-

sége az, hogy az eredeti levél fejrészébe írt adatoknak nem kell feltétlenül meggyezniük a boríték részben továbbítottakkal, a feldolgozás mégis a boríték adatai alapján megy végbe (a címzett például az lesz, akit a borítékban meghatároztunk, és nem feltétlenül az, aki a levélben szerepel).

Mint arra már utaltam, ez a boríték a felhasználó előtt az esetek többségében rejtve marad. Amivel mi, földi halandók találkozunk, dolgozunk, az az ábrán a Message Contents részben látható – maga az üzenet. A továbbiak során csak ennek a résznek a vizsgálatával foglalkozunk, a többi SMTP protokoll (RFC 821) feladata. Az üzenet két részből áll, ezek a fejlécek (Header) és az üzenet törzse (Body). Egy nagyon egyszerű levél valahogy így néz ki:

```
From: beno@netacademia.net
To: jeno@netacademia.net
Subject: Ez egy egyszeru level
```

Ez a level torzse.

Mindenekelőtt fontos megjegyezni, hogy a szabvány szerint a teljes kommunikáció ember által olvasható formában, mégpedig szigorúan a 7 bites ASCII karakterkészlet elemeit használva zajlik. (Ha ettől el akarunk térni, az üzenet tartalmát kódolni kell, de erről majd később.) Az üzenet fejrészét és törzsét egy üres sor választja el, ami a sorvégejelként használt „<CRLF>” karakterpár segítségével kifejezve „<CRLF><CRLF>”. Minden, ami a karakterozszozat előtt szerepel, fejlécnek tekinthető, és természetesen minden, ami ezután található az üzenetben, az maga az üzenet törzse. Az elválasztó üres sor nem tartozik az üzenethez. A törzs megadása nem kötelező, ekkor jutunk el az „üres üzenet” definíciójához :-)

### Fejlécmező

Az üzenet fejrészé egy vagy több mezőt (fejléct) tartalmaz (a fenti példa például rögtön hármat). Egy fejléc a következőképpen épül fel:

fejléc-név:fejléc-törzs<CRLF>

Egy fejléctörzs átlóghat a következő sorba, mégpedig úgy, hogy a törzsrészben található úgynevezett „whitespace” karakterek (például a tabulátor és szóköz) elé sorvégejel (<CRLF>) szúrunk. Ilyenkor a következő sor természetesen a whitespace karakterrel kezdődik, ebből lehet tudni, hogy az nem egy új fejlécmező nevének kezdete, hanem az előző mező törzsének folytatása. Például:

```
From: beno@netacademia.net
To: jeno@netacademia.net
Subject: Ez egy tobb sorban
        elkuludott fejrész, ami
        mégis egy sorban jelenik meg majd.
```



A fejléc csak ASCII karaktereket tartalmazhat, azok közül is csak a nyomtatható fájtákat (*plusz természetesen a whitespace karakterek: szökőz, tabulátor, sorvéjél*). A mezők neve és törzse értelemszerű okokból nem tartalmazhatja a kettőspontot.

Az RFC 822-ben definiált és ajánlott fontosabb mezők:

- ☞ From: Az üzenet feladója
- ☞ Sender: Az a személy, program, vagy szolgáltatás, aki/ami az üzenetet elküldte
- ☞ Reply-to: Az a cím, ahova a válaszüzeneteket várjuk (*nem feltétlenül egyezik meg a feladó címével*)
- ☞ To: Az üzenet címzettje(i) (*elsődleges címzettek*)
- ☞ Cc: Másolatot kapnak (*másodlagos címzettek*)
- ☞ Bcc: „Vakmásolatot” (*Blind Carbon Copy*) kapnak. Az itt felsorolt címek nem szerepelnek az üzenetben (*csak a borítékban*), tehát az üzenet többi címzettje nem szerez tudomást róluk (*és ők egymásról sem*).

Ezen kívül léteznek további, üzenetküldéshez használatos mezők (*Message-ID, Referencés, Keywords*), dátum és időazonosítók (*Date*), és természetesen az üzenet témája (*Subject*), valamint a fejrészen szerepelhet bármilyen, felhasználó által definiált mező (*X-.....*). A mezők sorrendje egyébként nem kötött.

**Érdeklösség, hogy a USENET hírcsatornák alapját is az SMTP levelek képezik. A legfontosabb különbség az, hogy ilyenkor a címzett To: mező helyett Newsgroups: szerepel, de az üzenetek formátuma teljesen megegyezik a hagyományos levelekkel.**

#### Dátum

A dátummezőben használt időformátum az alábbi:

Thu, 15 Mar 2001 17:53:27 -0800

A nap angol megnevezése (*és az azt követő vessző*) elhagyható, azután következnek a nap (*egy vagy két karakteren*), majd a hónap rövid angol neve, az év (*az szabvány szerint még kettő, Y2K óta természetesen négy karakteren*), az óra, perc és másodperc két-két karakteren, kettősponttal elválasztva (*a másodperc elhagyható*), végül az időzóna. Az időzóna lehet a Greenwich-i középideőtől való eltérés +/-óópp formátumban, valamelyik általános zónarövidítés (*pl. GMT, UT, EST, CST, stb.*), illetve érdekes módon amerikai katonai meghatározás (*pl. A-Z*).

#### A címzés formátuma

A fejlécmezők többsége természetesen e-mail címeket tartalmaz. A legegyszerűbb mód természetesen, ha csak az e-mail címet adjuk meg:

To: beno@netacademia.net

De a címkehez általában tartozik valamilyen becsületes név is, ilyenkor a nevet idézőjelbe tesszük, az e-mail címet pedig kacsacsőrök közé zárjuk:

To: "Fu Beno" <beno@netacademia.net>

A felhasználó „barát” levelezőprogramokban ilyenkor csak a „szép név” jelenik meg.

Több címet úgy adhatunk meg, ha a fenti formátumban el-

készített címeket egymás után vesszővel elválasztjuk. Természetesen ilyenkor is használhatjuk a mezők több sorba történését (*ezt egyébként a szabvány „folding”-nak hívja*):

To: "Fu Beno" <beno@netacademia.net>,  
"Fenyó Jeno" <jeno@netacademia.net>,  
Telapo@funet.fi

#### Ennyi?

Bizony, körülbelül ennyi. Habár nem tértünk ki arra, hogy milyen fejléc-mezőket definiáltak például az üzenetek továbbításához, visszaküldéséhez, az RFC-822 legfontosabb részeit megismertük. A levelezés akkor még kizárólag ASCII szöveges üzenet továbbításáról szól, szó sem volt alternatív tartalomról, karakterkészletekről, csatolt fájlokról.

#### UUEncode/UUDecode

Miután a felhasználók kezdtek rászokni a hálózaton keresztüli levelezésre, egyre nagyobb volt az igény arra, hogy a szöveges levelekbe szövegben kívül mást is tölthessenek. Az egyik legelső ilyen módszer a UUEncode (*Unix-to-Unix Encode*) volt, amit a USENET hírcsatornákon egyébként a mai napig gyakran használnak.

Az üzenetek formátuma nem változott: a csatolt állományokat az üzenetek törzsébe, annak is a végéhez csatolják, egy speciális fejléc után. A kódolás során a bináris adatot valahogyan az üzenet törzsébe tuszoklják úgy, hogy csak szabványos, 7 bites ASCII karaktereket használunk. A trükk: az üzenet minden 3 bajtját négy hatbites darabra bontanak, majd az így kapott értékekhez hozzáadnak 32-t. A bajtok értéke 32 (*szökőz*) és 32+(2<sup>6</sup>-1)=95 (*alulvonal*) közé kerül, és könnyen beilleszthető az üzenetbe (*cserébe a csatolt állomány mérete megnő, mégpedig 4/3-ára*.)

From: "Zxxxxx" <zxxxxx@bikerider.com>  
Newsgroups: alt.binaries.pictures.motorcyc  
Subject: Re: Yamaha de MX roya y blanca  
Lines: 1153  
Date: Thu, 15 Mar 2001 08:09:33 GMT

Parece un salto de la h\*stia, pero es porque la calle está más baja que el circuito, de ahí la altura de las farolas :-)

Vs  
Xavi

```
begin 666 vz125.jpg
^O2D9)1e^1^0s Ee^6^`^VP!#^`^"e,"^"e,#^P,$^P,$!0e%
M!OH^!P8(# H,# L*"PL-#A(0#X1#@L+!$180$!,4#145# ^7k
[... ]
;#045#Z6GD"">U,<E1!&RS9LPA0#^,0&K5:2E(\ZXZ4N&10
1^!PO:JKVE?P?>1Y^!1_]D^
end
```

Fülöp Miklós

[[mik@netacademia.net](mailto:mik@netacademia.net)]

#### A cikkben szereplő URL-ek:

- [1] <http://www.ietf.org/rfc/rfc0822.txt>
- [2] <http://www.ietf.org/rfc/rfc0821.txt>



# Elektronikus kereskedelem

Most, amikor Magyarországon az Országgyűlés már tárgyalja az elektronikus kereskedelemről szóló törvény tervezetét, és az előzetes információk szerint májusban várható a törvény elfogadása, talán nem szükségtelen, hogy a magyar jogalkotásra kétségtelenül legnagyobb befolyást gyakorló, és a jogharmonizációs kötelezettségek miatt a kereteket meghatározó szabályozást, az Európai Unióban e tárgyban megszületett irányelvet részletesebben megvizsgáljuk.

A szabályozás irányelv formájában jelent meg, amelyet az 1998.-as első tervezetet követően az EU döntéshozatal-előkészítési folyamatának számos lépéscsjét megjárva, néhány módosítás után 2000. június 8-án fogadtak el. Az irányelv, mint jogforrás a tagállamokra további kötelezettséget ró, annak keretei között a meghatározott határidőn belül meg kell alkotni a nemzeti jogszabályt. Így tehát az EU tagállamokban lázas munka kezdődött az e-commerce törvényi szabályozására.

Az elektronikus kereskedelemről szóló irányelv kizárólag az internal market, az Unió's belső piac jogi viszonyainak rendezésére vállalkozik, külső (external) vonatkozásokkal nem foglalkozik. A nem uniós országokban működő szolgáltatókat tehát azok a lehetőségek, amelyeket az Irányelv kínál, nem illetik meg, s ha erre mégis igényt tartanak, akkor valamely uniós országban regisztrálniuk kell magukat.

A szabályozást öt lényeges kérdés köré csoportosították:

**Az információs társadalomban a szolgáltatói szféra megalapozása (establishment of service providers).** Az Irányelv célja, hogy e téren a jogi bizonytalanságot eloszlassa. Két alapelvet mond ki e körben: a service provider-jellegű szolgáltatást nem lehet különleges engedélyezéshez kötni, ugyanakkor az SP köteles magáról – pontosan meghatározott, azonosításához szükséges – adatokat közölni.

**A kereskedelmi kommunikáció (reklám, direkt marketing stb.).** Az alappondatok: a kommunikáció megbízhatósága (a visszaélések veszélyének csökkentése, felhasználó-fogyasztóvédelem), valamint az egyes különleges hivatások (jogász, orvos) sajátos – s a szakmai etikával összeegyeztethető – reklámjának megengedése.

**Az online szerződéskötés joghatályának elismerése.** Az alapelv e körben: a szerződéskötési szabadság érvényre juttatása mellett a kormányoknak olyan jogi környezetet kell teremteni, mely az elektronikus szerződéskötés feltételeit és joghatályát biztosítja.

**A közreműködők felelősségi kérdéseinek rendezése.** Az alapelv: egyrészt a felelősségi rendszer (felelősségi lánc) felépítése, másrészt ezen belül az ISP felelősségének korlátozása a közreműködők, azaz szolgáltatók igénybe véve szolgáltatók cselekedeteiért.

**Végrehajtás.** E körben a tagállam felé alapkövetelményként fogalmazza meg az Irányelv az online környezet gyors, hatékony jogi rendezésében való közreműködést, új a nemzetközi szabályozási munkában, mint a nemzeti szintű jogalkotásban.

Az Irányelv a szankciók megállapítását a nemzeti jogalkotásra bízta azzal a kritériummal, hogy a szankcióknak haté-

konyknak, arányosnak és visszatartónak kell lennie.

Nem tartoznak az Irányelv hatálya alá az adózási kérdések, a 95/46/EC és a 97/66/EC Irányelvekben meghatározott információs társadalmi szolgáltatásokkal kapcsolatos kérdések, a kartelljog által szabályozott területek, az információs társadalmi szolgáltatások közül a közjegyzők, vagy velük azonosnak minősülő szakmák, amelyek közvetlen és speciális kapcsolatot igényelnek, és közhatalmat gyakorolnak, az ügyfél vagy védelem képviselője a bíróság előtt, és a szerencsejátékok.

Az Irányelv az elektronikus kereskedelem tárgya és szereplői szempontjából az alábbi fogalmak meghatározását tartotta fontosnak.

## Információs társadalmi szolgáltatás

Minden olyan szolgáltatást ide sorol, amelyet általában díjazás ellenében, távolról, elektronikus úton és a szolgáltatás címzettje egyéni kérésére nyújtanak. Ide tartozik tehát bármilyen B2B vagy B2C viszonylatban általában ellenszolgáltatás fejében, de kivételesen ingyenesen is (olyan honlap estében, amelyre a látogató az Internet-hozzáféréseért fizetendő díjtól eltekintve ingyen vehet igénybe, a szolgáltatást a beépített reklámok tartják el) nyújtott szolgáltatás: pl. Internet-hozzáférés biztosítása, weboldal fenntartása (tárhelybiztosítás), Bulletin Board Service (BBS, elektronikus faliújság) fenntartás, e-mail szolgáltatás, kereső- és portálszolgáltatás, online termékkértékesítés, más online szolgáltatások (pl. pénzügyi) nyújtása, online kiadó (újság), online kulturális szolgáltatás, különösen szerzői művek felhasználása (lehívása letöltési lehetőséggel vagy anélkül). Lényeges tartalmi elem, hogy a szolgáltatás nyújtása telekommunikációs eszköz útján, elektronikus úton történjen a szolgáltatást igénybe vevő írásbeli vagy áutaló magatartással kifejezett (rákattintás - click on) megrendelése alapján. A felek távol vannak, azaz nincsenek együttesen, adott helyszínen jelen a szerződés megkötésekor. A szolgáltatásnak minden esetben egyedileg igényelnie kell lennie, olyannak, amit adatok egyéni kérésre történő sugárzásán keresztül nyújtanak.

## Szolgáltató

Minden olyan természetes, vagy jogi személy, amely információs társadalmi szolgáltatást nyújt. E tekintetben szolgáltató az access provider (a hálózati távközlési hozzáférést biztosító szolgáltató pl. Matáv, Westel, stb.), a Service Provider (a szűk értelemben vett Internetszolgáltató, aki a szerverén adatközlést, honlap-fenntartást, a hálózaton adatait továbbítást és egyéb szolgáltatást is nyújtó szolgáltató), a content provider tartalomszolgáltató, akinek honlapja útján az Internet felhasználói, és viszonylatban fogyasztók a tartalomszolgáltatónak az előbbieken említett szolgáltatók közreműködésével nyújtott szolgáltatásait elektronikus kommunikáció útján igénybe veszik, valamint az Internetszolgáltató és tartalomszolgáltató együttes szerepét ellátó online szolgáltató.



tó, aki a tartalomszolgáltatás mellett tárhelyet is biztosít, pl. BBS elektronikus faliújság útján.

Megalakult szolgáltatónak nevezi az Irányelv az olyan szolgáltatót, aki a gyakorlatban gazdasági tevékenységet végez, állandó telephelyet használva határozatlan időre. A szolgáltatás nyújtásához szükséges műszaki eszközök és technológiák jelenléte és használata nem jelenti azt, hogy a szolgáltató megalakult.

Az információs társadalmi szolgáltatóvá válást nem lehet előzetes engedélyezéshez, vagy bármilyen más olyan akushoz köti, mely hatósági hozzájárulás formájában nyilvánul meg. Ez azonban magához az ISP-jelleghöz kötődő különleges engedélyekre vonatkozik csak, egyébként az általános kereskedelmi szereplővé válási feltételeknek meg kell felelniük.

A szolgáltató – minden szolgáltató – köteles magáról széles körben információt nyújtani. Neve, működése szerinti székhelye, kapcsolattartás a szolgáltatóval e-mail formában, hol, mely kereskedelmi nyilvántartásban regisztráltak, mi a regisztrációs száma, engedélyhez kötött tevékenység esetén az engedély száma, annak a szakmai szervezetnek megnevezése, melyhez – ha az országban van ilyen – a szolgáltató tartozik, s végül – ha a tevékenység forgalmi adó köteles, az ország saját szabályai szerinti adóazonosító vagy adószám megadása.

A kereskedelmi szolgáltatókkal szemben az Irányelv többkövetelményeket állít. Első a „tisztán azonosíthatóság”, második a felelős személy megnevezése, harmadik a promóciós-reklámjellegű anyagok (*akciók, ajándékok stb.*) sajátos megjelölése, a negyedik a kereskedelmi vetélkedők és játékok engedélyhez kötési lehetőségének elismerése mellett az ilyen akciók egyértelműségének, tisztességének biztosítása.

### A szolgáltatás címzettje

Minden olyan természetes, vagy jogi személy, aki szakmai célból, vagy más okból információs társadalmi szolgáltatást vesz igénybe, különösen azért, hogy információhoz jusson, vagy azt elérhetővé tegye.

### Fogyasztó

Fogyasztó alatt a már hagyományosan elfogadott fogalommal találkozhatunk ismét: bármely olyan természetes személy, aki a saját üzleti, kereskedelmi vagy szakmai tevékenysége körén kívül eső cél érdekében cselekszik.

Az Irányelvben önálló alfejezetet kapott a kereskedelmi kommunikáció (*Commercial Communication*). Ez három körből áll: a szolgáltatók által nyújtott többletkövetelmények, a nem kívánt kereskedelmi e-mail kérdése, s egyes, külön szabályozás alá eső hivatások reklámozásának kérdése.

Kereskedelmi közleménynek minősül a közlés bármilyen formája, ami árucikkeknek, szolgáltatásoknak, továbbá cég, szervezet, illetve kereskedelmi, ipari, vagy kézműves tevékenységet folytató, illetve szabadfoglalkozású személy arculatának közvetett, vagy közvetlen promóciójára szolgál. Nem minősülnek kereskedelmi közleménynek egy vállalat, szervezet, vagy személy tevékenységéhez való közvetlen hozzáférést lehetővé tevő információk, így különösen domain név, vagy e-mail cím továbbá az árucikkekre, szolgáltatásokra, vagy cég, szervezet, illetve személy arculatára vonatkozó, függetlenül, így különösen nem anyagi megfontolásból összeállított közlemény.

A kereskedelmi közlemény kötelező tartalmi elemei a következők:

- ☞ a kereskedelmi közlemény legyen egyértelműen azonosítható
- ☞ egyértelműen legyen azonosítható az a természetes, vagy jogi személy, akinek érdekében a kereskedelmi közleményt megalkották
- ☞ a promóciós ajánlatok, például kedvezmények, jutalmak és ajándékok (*amennyiben engedélyezettek*), egyértelműen azonosíthatók legyenek, az azok igénybevételéhez előírt feltételek pedig könnyen hozzáférhetően, pontosan és egyértelműen legyenek feltüntetve
- ☞ a promóciós versenyek, vagy játékok (*amennyiben engedélyezettek*), egyértelműen azonosíthatók legyenek, és a részvételhez előírt feltételek könnyen hozzáférhetően, pontosan és egyértelműen legyenek feltüntetve.

Külön foglalkozik az Irányelv a nem igényelt kereskedelmi közlemények (*spam*) kérdésével is. E tekintetben a tagállamokra bízta annak szabályozását, hogy a nem igényelt, e-mailen terjesztett kereskedelmi közlemények világosan, és egyértelműen azonosíthatóknak kell lennie a címzett általi átvétel időpontjában. A spam körében új rendelkezés az opt-out nyilvántartások felállításának igen határozott támogatása: a természetes személyek számára biztosítani kell a jogot, hogy e nyilvántartásba felvételt nyerhessenek, s így a nem kívánt kereskedelmi kommunikáció ellen védekezhesnek.

Az alkalmazandó jogként az Irányelv annak az országnak a jogát tartja irányadónak, ahol a szolgáltatást nyújtó székhelye van. A Section 3. az elektronikus szerződésekéről rendelkezik. Itt is rögzíti az Irányelv, melyek a kötelező elemek, s milyen feltételek mellett lehet a szerződési folyamatot biztosítani. Négy körben jelöli meg azokat a kivételeket, mikor nem lehet elektronikus szerződési formát választani:

- ☞ minden olyan szerződés, mely ingatlanra vonatkozó jog átszállásával kapcsolatos (*kivéve a bérleti jogot*)
  - ☞ olyan szerződések, melyeknek érvényességéhez valamely közhitelű nyilvántartásban való regisztráció szükséges
  - ☞ kezesség, jótállás, garancianyújtás olyan esetben, ha a cél az adott személy üzleti, kereskedelmi vagy szakmai körén kívül esik
  - ☞ a családjog és az öröklési jog körébe tartozó szerződések
- Az Irányelv azt is előírja, hogy a szolgáltatóknak az elektronikus szerződéskötési folyamatban milyen tájékoztatósi, adatrögzítési, és hozzáférés-biztosítási kötelezettségei vannak. Az egyik legvitatottabb kérdést, a szerződés létrejöttének időpontját az Irányelv akként rendezi, hogy nem elegendő a szerződés létrejöttéhez az egyszerű klickekés, hanem szükséges a szolgáltatótól való visszajelzés a megrendelés átvételének elismeréséről, továbbá a megrendelőtől annak visszaigazolása, hogy az elismervényt megkapta. E körben két további elvet kell alkalmazni: a rendelést akkor lehet elfogadottnak mikor, a szolgáltatás igénybe vevője ahhoz hozzá tud férni. Ugyanakkor a szolgáltatóknak kötelezettsége a rendelés visszaigazolásának haladéktalan elküldése. További előírás, hogy a szolgáltatóknak a szerződési- és általános feltételeket úgy kell hozzáférhetővé tenni, hogy a vevő ezt rögzíteni (*menteni*) és reprodukálni tudja.

A másik legtöbbször vitát kiváltó kérdés: a Service Provider (*Inter-net-szolgáltató*) felelőssége. Az Irányelv a közreműködő Inter-



netzszolgáltató felelősségét korlátozza, és a szolgáltatói felelősséget kizárja azokban az esetekben, mikor nem ő kezdeményezte az adatátvitelt, nem ő választotta meg az adatátvitel címzettjét, s nem ő határozta meg az adatátvitel tartalmát.

A caching-ért, és a hosting-ért való szolgáltatói felelősség kizárásának feltételeit is az ISP előbb vázolt felelősségéhez hasonlóan rendezi az Irányelv. A szolgáltatók szempontjából lényeges annak kimondása, hogy nem terheli őket kötelezettség az általuk továbbított vagy tárolt információ figyelésére, nincs tehát monitoring-kényszer, ez azonban nem érinti a hatóságok törvényi felhatalmazás alapján történő hasonló törekvéseinek semminemű korlátozását.

Az irányelv a viták rendezését lehetőleg a bíróságokon kívüli fórumok kezébe kívánja helyezni hasonlóan a fogyasztóvédelem területén mutatkozó megoldásokhoz. Optimálisabb megoldást jelenthetne a mindenhol igen lassan működő igazságszolgáltatással szemben a gyors, hatékony, egy-, legfeljebb kétfokú, hozzáértőkből álló vitarendezési fórumok felállítása, és az Internettel kapcsolatos vitáknak itt történő rendezése. Az Internet nem tűri a lassú döntéseket: mi-

re a bírósági határozat megszületik, a peres felek érdekei már régen meghaladtakká válnak. A speciális technikai háttér megfelelő szintű ismerete is szükségessé válhat a jogvita alapos megismeréséhez és eldöntéséhez. A domain nevekkel kapcsolatos vitarendezési eljárások (WIPO illetve a magyar megoldás Tanácsadó Testület illetve eseti választott bíróság) már bizonyítottak, működnek. Az elektronikus kereskedelemmel kapcsolatos viták területén is minden bizonnyal hatékony alternatívája lehet a hagyományos bírói útnak egy másik fórum.

Dr. Mayer Erika  
Nádas & Mayer Ügyvédi Iroda  
1137 Budapest, Radnóti Miklós u. 38.  
Tel: 340-4150, 06-309-318-089  
E-mail: mayer@nadas-mayer.hu



## 64 és 128 Kbit/sec-os

# Bérelt vonalak a távközlési szolgáltató díjával



- ✓ belépési díj nélkül
- ✓ router biztosításával
- ✓ webservert működtetésével
- ✓ ajándék telefonos interneteléréssel
- ✓ 1 db .hu és 1 db .com domainnév regisztrációjával
- ✓ DNS szerviz biztosítással

64k: 85.000,-  
128k: 129.000,-

Az áraink ÁFA nélkül és 2001. április 30-ig érvényesek.  
Érdeklődjön a 06-40-HUNNET telefonszámon vagy info@ahol.com címen!

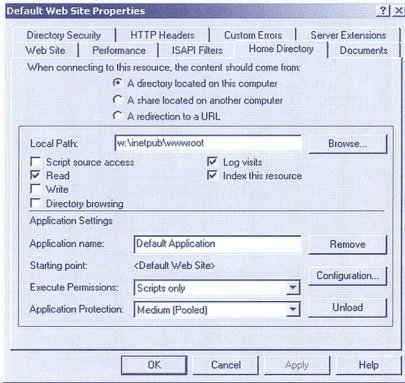


# ASP sulis (IV. rész)

A kis nyelvészkedési kirándulás után hamarosan visszatérünk az ASP programozáshoz. E számunkban bemutatjuk az Application és Server objektumokat, ennek kapcsán kitérünk az ASP alkalmazások mibenlétére is, végül egy újabb naplózási trükk következik majd. Mint mindig, a példaprogramok megtalálhatók az [1] címen.

## Az ASP alkalmazás

Mint arra már néhány héttel ezelőtt kitértünk, az ASP alkalmazás egy adott virtuális könyvtárban (és annak *gyermekkönyvtáraiban*) elhelyezett ASP fájlok gyűjteménye, valamint természetesen az azokat körülvevő keretrendszer. Az IIS már telepítésekor létrehozta a teljes webet magába foglaló alapértelmezett ASP alkalmazást, a „Default Application”-t. Nyissuk meg a Default Web Site tulajdonságlapját, és kattintsunk a Home Directory (alsóbb könyvtárszinteken csak „Directory”) oldalra:

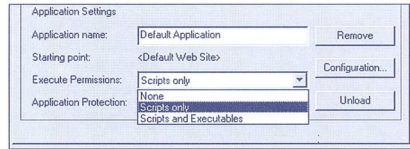


➤ Az alapértelmezett webalkalmazás beállításai (az ablak első felében láthatók)

Az Application Settings választóvonal alatt található elemek mind-mind az ASP alkalmazás kezelésére szolgálnak. Mindegyiket láthatjuk az alkalmazás nevét, a kezdőpontját (*gyökérkönyvtárt, esetünkben ez maga a virtuális web*), valamint a scriptek futtatására és az alkalmazás védelmére vonatkozó ablakokat. Mielőtt ezek értelmezésébe belemennénk, lássuk a gombokat: a Remove eltávolítja, megszünteti a webalkalmazást (a lemezen található fájlok ettől nem sérülnek meg). Bár a gyökéralkalmazás is megszüntethető, ennek véghezvitele csinos kis hibázenetekhez vezet mind a böngészőben, mind a kiszolgáló eseménynaplójában. A hierarchia alsóbb szintjein található alkalmazásokat viszont különösebb probléma nélkül megszüntethetjük.

Az Unload gomb megnyomására a webalkalmazás kitöltődik, távozik a memóriából. Törölődnek az ASP objektumok adatai, felszabadulnak az IIS által fogott DLL-ek, és a legközelebbi alka-

lommal ismét futnak a definiált inicializáló rutinok. A Configuration gomb megnyomására megjelenő párbeszédablak már ismerős lehet, és még lesz is szó róla: ebben lehet meghatározni az egyes kiterjesztésű fájlok kezelő rendszerkomponenseket, a hibakeresés (ASP debugging) beállításait, valamint az alkalmazás globális paramétereit (alapértelmezett scriptnyelv, időtűlépés, stb.)



➤ A választható futtatási jogosultságok

## A futtatási jogosultságok

Az „Execute Permissions” mezőben három lehetőség közül választhatunk:

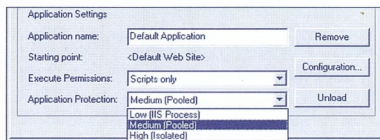
- None: Csak statikus működés engedélyezett. Scriptek és programok nem futtathatók. Ezt használjuk akkor, ha nem használunk aktív komponenseket
- Scripts only: Scriptfájlok végrehajtása engedélyezett, de különálló programok továbbra sem futtathatók. Ez az alapértelmezett és ajánlott beállítás
- Scripts and Executables: A legveszélyesebb, CGI-kompatibilitás miatt megtartott opció. Ilyenkor a scriptek mellett a könyvtárakban található programok is lefuttathatók.

## Az alkalmazás védelme

A meghatározás tulajdonképpen hibás, ugyanis nem a webalkalmazásunkat, hanem magát az IIS-t védjük a saját csinytevéseinktől (na jó, esetleg magunkat a többi webalkalmazástól). Hogy miért kell ez a védekezés?

A webalkalmazások saját útjait járják. Az esetek többségében (például .dll fájlokban megírt) külső komponenseket használunk, hibátlan program pedig, mint tudjuk, nincsen. A .dll-ek közismert tulajdonsága, hogy a szülőprocessz címtartományába töltődnek be, és ha a .dll készülő elhalálozni, a szülőprocessz is magával rántja. Így lehet egyetlen .dll segítségével a nullával egyenlővé tenni a webkiszolgálót.

Pontosan az ilyen esetek elkerülésére találták ki az „alkalmazások védelme” („Application Protection”) opciót. Egy (.dll-eket töltőgető) webalkalmazás (kvázi maga a .dll) három módon töltődhet be:



➤ A webalkalmazás három védelmi szintje

- ☞ (Low): az IIS processzbe. Ez gyors, de veszélyes működést eredményez, hiszen ha a .dll kipukkan, meg vele az IIS is.
- ☞ (Medium – Pooled): egy, az IIS processzétől elválasztva futó processzbe. A processz neve dllhost.exe, megleshetjük a Task Managerben. Minden Medium szintű webalkalmazást ugyanaz a dllhost.exe futtat, tehát ha a több Medium szintű alkalmazásból egy elhasal, magával rántja a többit is – az IIS viszont talpon marad. Ez az alapértelmezés, mert viszonylag biztonságos, és nem igényel sok erőforrást.
- ☞ (High – Isolated) A High szintre helyezett webalkalmazások mindegyike saját dllhost.exe-t kap, amibe annyira és annyiszor küvül bele, ahányszor csak akar, önmagán kívül senkinek sem árthat vele. Akit érdekel, kipróbálhatja: nyisson meg egy Task Manager-t, állítsa a futó processzeket név szerint sorba, és meglátja, hogy n darab dllhost.exe fut. Majd állítson egy webalkalmazást High szintre, és nyisson meg egy oldalt belőle: a futó dllhost.exe-k száma n+1-re nő (az Unload hatására pedig értelemszerűen eggyel csökken). Ez a legbiztonságosabb megoldás, de a gyakori kontextusváltás miatt sok erőforrást igényel, ezért a Microsoft nem ajánlja, hogy egy kiszolgálóra 2-3-nál több ilyen szintű alkalmazás kerüljön.

### Saját webalkalmazás létrehozása

Saját webalkalmazást úgy hozhatunk létre, hogy egy szimpatikus könyvtár tulajdonságlapján megnyomjuk a Create... gombot. Ekkor ez a könyvtár lesz az újonnan létrehozott webalkalmazásunk gyökérkönyvtára, aminek különleges jelentősége van: azok az ASP fájlok osztanak meg ugyanis közös ASP objektumokat (például az *Application* objektumot), amelyek ugyanabban a webalkalmazásban találhatók.

Az *Application* objektum tehát nem más, mint egy közös tárolóhely, amit az adott webalkalmazás minden scriptje elér. Az *Application* objektumban adatokat, objektumokat tárolhatunk el, hogy majd később kiolvassuk onnan (*csakúgy, mint a Session objektumnál*):

```
Application("counter") = 12
Set Application("myObject") = objObject
Set objObject2 = Application("myObject")
```

Amint az a fenti példában is látható, az *Application* objektumban is tárolhatunk más objektumokat. Nem szabad elfelejtenünk azonban két nagyon fontos dolgot:

- ☞ Az objektumokat Set értékadás segítségével kell az *Application* objektumba betölteni, és a kiolvasásuk is a Set parancs segítségével történik.
- ☞ Nem minden objektum alkalmas arra, hogy az *Application* objektumba töltsük! Az IIS ugyanis többszálú alkalmazás, egyidőben több felhasználót szolgál ki. Az *Application* objektumba csak olyan objektumokat tölthetünk be, amelyek többszálú (*FreeThreaded*) végrehajtási módban képesek működni és úgynevezett Both threading-modell alkalmaznak. Lásd még: [2]

Az alábbi példa az *Application* objektumot használja egy primitív számláló létrehozásához

```
<%
Application.Lock
Application("count") = Application("count") + 1
Application.Unlock

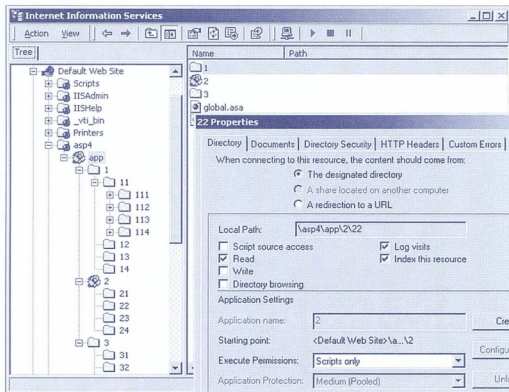
Response.Write("Üdv! Te vagy az IIS újraindítása óta a(z) " & Application("counter") &
látogató.")
%>
```

A példa a valóságban azért használhatatlan, mert a webalkalmazás újraindulásának pillanatában – reboot vagy unload esetén – az *Application* objektum tartalma, ezzel pedig a számláló értéke is elveszik. De nem is ez a célunk vele, hanem a demonstráció. Az első és a harmadik sorban látható *Application.Lock()* és *Application.Unlock()* metódus használatára azért van szükség, mert az *Application* globális objektum, és előfordulhat, hogy egyszerre több felhasználó szeretné írni ugyanazokat az adatokat. Az ütközések elkerülése végett minden írásművelet előtt az objektumot zárolni kell, majd a lehető legrövidebb időn belül az *Unlock()* metódus segítségével fel kell oldani, ugyanis míg az *Application* objektum zárolva van, senki más nem férhet hozzá, mint az, aki eredetileg zárolta azt.

**Próbáljuk ki!** A *lock1.asp* zárolja és néhány másodpercig zárolva tartja az *Application* objektumot. Ha a *lock1.asp* futásának ideje alatt megnyitjuk a *lock2.asp* oldalt, az csak a *lock1.asp* futásának befejezése után lesz képes hozzáférni az *Application* objektumhoz.

Ha mi nem oldanánk fel a zárolást, az IIS az oldal végrehajtása után, de legkésőbb a scriptfuttatás időtűlépésekor felszabadítja azt.

Természetesen az *Application* objektum tartalmát is elérhetjük



### ☞ Webalkalmazások hierarchiája

A fenti ábrán látható hierarchiában most figyeljük az *app* és a 2 könyvtárakat (ezek webalkalmazások). Ha megnyitjuk a hierarchiában alattuk található könyvtárak tulajdonságlapjait, láthatjuk, hogy melyik webalkalmazáshoz tartoznak: 22 nevű alkönyvtár például a 2-höz, míg a 112 az *app*-hoz, azaz mindegyik ahhoz legközelebbi ősehez, ami webalkalmazás gyökérének funkcionál.

### Az *Application* objektum





kollekciókon keresztül, erre szolgál az `Application.Contents` kollekció (`contents.asp`):

```
For Each oItem In Application.Contents
    Response.Write(Application(" " & oItem & ") = " &
    Application(oItem) & "<br>")
Next
```

Ebből a kollekcióból is pontosan ugyanúgy lehet törölni elemeket, mint a Session-nál is láttuk:

```
Application.Contents.Remove("counter")
Application.Contents.Remove(2)
Application.Contents.RemoveAll()
```

Az első sor a "counter" értékét, a második sor az Application objektumon belüli második változót, míg a legutolsó sor az Application objektum teljes tartalmát törölte.

Az `Application.StaticObjects` kollekció a `global.asa` fájlban az <OBJECT> elem segítségével, „Application” scope-ban létrehozott változókat tartalmazza. A kollekció pontosan úgy használható, mint az `Application.Contents`:

```
For Each oItem In Application.StaticObjects
    Response.Write(Application(" " & oItem & ") = " &
    Application(oItem) & "<br>")
Next
```

Az egyetlen különbség az, hogy ebből a kollekcióból nem törölhetünk elemeket (azok elvesznek maguktól a webalkalmazás újraindításakor).

### A global.asa fájl

Mint azt már a Session objektum ismertetésekor röviden leírtam, a `global.asa` fájl egy speciális állomány, amit az ASP alkalmazás gyökérkönyvtárában kell elhelyezni (de a használata nem kötelező). A fájl arra való, hogy ebben helyezzük el a globális objektumokat létrehozó és eseményeket kezelő kódreszleteket, úgyis mint:

- az Application objektum eseményeit (létrehozását, megsemmisítését) kezelő rutinok (`Application_OnStart` és `Application_OnEnd`)
- a Session objektumok létrehozását és megsemmisítését kezelő eljárások (`Session_OnStart` és `Session_OnEnd`)
- globális objektumok létrehozása az <OBJECT> elem segítségével
- típuskönyvtárak (*type libraries*) betöltése

A `global.asa` fájlban található rutinokat <SCRIPT></SCRIPT> elemek közé kell elhelyezni. A statikus objektumok létrehozására szolgáló <OBJECT> elemet a <SCRIPT> blokkon kívülre kell elhelyezni, míg a típuskönyvtár-definíciók a blokkon kívülre és belüire egyaránt kerülhetnek, de érdemes azt is már a fájl elején, a <SCRIPT> blokkon kívül letudni.

Ha a `global.asa` fájl tartalma megváltozik, az IIS minden már megkezdett kapcsolatot kiszolgál, és csak azok lezárása után tölti be az új változatot. Ez természetesen az összes Session és az Application objektum lezárását és újbóli megnyitását is jelenti. A `global.asa` betöltése és feldolgozása során az IIS nem fogad új kapcsolatokat, a felhasználók ezidő alatt csinos kis hibaüzenettel találkozhatnak

(„a kérés nem szolgálható ki, az alkalmazás újraindul”).

### Események a global.asa-ban

A `global.asa`-ban négy különféle eseményt kezelő rutint definiálhatunk, ezek az `Application_OnStart`, `Application_OnEnd`, `Session_OnStart`, és a `Session_OnEnd`. Lássunk egy mintát:

```
<SCRIPT LANGUAGE='VBScript' RUNAT='Server'>
```

```
Sub Application_OnStart
    Application("appstarttime") = Now
End Sub
```

```
Sub Application_OnEnd
    ...
End Sub
```

```
Sub Session_OnStart
    Session("starttime") = Now
    Set Session("oFS") =
    Server.CreateObject("Scripting.
    FileSystemObject")
End Sub
```

```
Sub Session_OnEnd
    Set Session("oFS") = Nothing
End Sub
```

```
</SCRIPT>
```

- `Application_OnStart`: a webalkalmazás indulásakor, egyszeri alkalommal fut le.
  - `Application_OnEnd`: az alkalmazás leállításkor fut le, az alkalmazás életében ugyancsak egyszer
  - `Session_OnStart`: Session objektum létrehozásakor, gyakorlatilag minden új felhasználó belépésekor hajtódik végre
  - `Session_OnEnd`: ez pedig a Session lezárásakor lép működésbe. (például ha időtűlépés vagy `Session.Abandon()` hívása miatt az IIS megszünteti a session-t)
- Nézzünk rá egy pillanat erejéig az `Application_OnStart` rutinra: írunk az `Application` objektumba, és mégsem zárjuk előtte? Nem lesz ebből baj? Nem, ugyanis ez az esemény egyszer és csak egyszer következik be, mégpedig az alkalmazás életartama legelejein, amikor másnak még esélye sincs az Application objektumhoz hozzáférni. Ez az egyetlen hely, ahol nem kötelező használni a Lock/Unlock metódust.

### Objektumok és típuskönyvtárak

Az objektumok létrehozásának módját a Session leírásánál már megismertük:

```
<OBJECT RUNAT=Server SCOPE=Application ID="oGFSO"
    PROGID="Scripting.FileSystemObject">
```

A különbség csak annyi, hogy a scope (*futási, létrehozási környezet*) most nem Session, hanem Application lesz. Ezután az oGFSO objektumot a webalkalmazás minden scriptjében közvetlen hivatkozással elérhetjük, és az megjelenik az `Application.StaticObjects` kollekcióban is. Ismét elmondom, hogy nagyon kell ügyelni arra, hogy az `Application` objektum-

ban csak Both, free threaded objektumok tároljunk (*ha <OBJECT> segítségével szeretnénk rossz objektumot létrehozni, az IIS hibát jelez, a dinamikus létrehozásnál viszont nem, ott csak később, a fura működés során derülhet fény a problémára*).

A COM objektumok általában típuskönyvtárakat is tartalmaznak. Ezekből a típuskönyvtárakból lehet kiolvasni az objektum metódusainak, jellemzőinek a listáját, de sokszor mindenféle konstansokat is.

Ha például egy fájl meg szeretnénk nyitni olvasásra a FileSystemObject segítségével, a következőt kell írunk:

```
Set oFSO = Server.CreateObject(
    ↳ "Scripting.FileSystemObject")
Set oFile = oFSO.OpenTextFile(„file.txt”, 1)
```

Ehhez tudnunk kell, hogy az OpenTextFile metódus második paraméterének 1 értéke az olvasást jelenti (*ForReading*). Ha azonban a global.asa tetején megadjuk a következő definíciót:

```
<!--METADATA TYPE="TypeLib" FILE="scrrun.dll"-->
```

(az scrrun.dll tartalmazza többek között a FileSystemObject objektumot), akkor így is írhatjuk:

```
Set oFSO = Server.CreateObject(
    ↳ "Scripting.FileSystemObject")
Set oFile = oFSO.OpenTextFile("file.txt",
    ↳ ForReading)
```

A számérték helyett tehát használhatók a típuskönyvtárban definiált konstansok, amelyek sokszor (*nagyon sokszor*) megkönnyítik az ember munkáját.

### A Server objektum

A Server objektum is egy és oszthatatlan, és főleg kényelmi szolgáltatásai miatt hasznos. Gondoljunk-e már például arra, hogy ezt írjuk ki a felhasználó böngészőjébe: „Vízszintes vonal: <HR>”. Ha ezt elküldjük a böngészőbe, megjelenik a felirat, majd maga a vízszintes vonal, hiszen a böngésző értelmezi a kódban található HTML tagot. Ha a HTML elemet magát szeretnénk megjeleníteni, a < jelet &lt; és >-t pedig &gt; entity-vel kell helyettesítenünk, tehát valahogy így: „Vízszintes vonal: &lt;HR&gt;”. Ezen kívül még sok más elemet is kódolni kell, nem beszélve a speciális karakterekről. Szerencsére itt van a [Server.HTMLEncode\(\)](#) metódus, ami elvégzi ezt a kódolást (*htmlenc.asp*):

```
Response.Write( Server.HTMLEncode(„Nesse <BR>” ) )
```

**Házi feladat:** Miért kellett a *htmlenc.asp* kódjában a *Server.HTMLEncode()* metódust duplán használni?

A HTMLEncode() azért is nagyon fontos, mert okos használatával elkerülhetjük az úgynevezett Cross Site Scripting [3] hibát. Ezt a fontos biztonsági hibát az okozza, hogy egyes weboldalak (akár hibáuzenett formájában is) válogatás nélkül, és kódolatlanul visszaküldenek bizonyos, részükre előzőleg átadott szövegrészeket. A *crosssite.asp* például bekéri a felhasználó nevét, majd kiírja a vonal alá. Írjuk be névnek az, hogy: `<script> alert(„Most formázom a winchesteredet!”); </script>`

Mi történik? A buta kód simán visszaküldi, amit kapott, bele a HTML kódba. A baj azért nagy, mert a gyánútlan felhasználót bárki egy megfelelően előkészített URL segítségével egy ugyancsak gyánútlan webkiszolgálóra irányíthatja, és már fut is a nem várt script.

A másik hasznos kódoló metódus a [Server.URLEncode\(\)](#). Mint azt talán mindenki tudja, az URL-ekben nem szerepelhet akár-milyen karakter. Ami nem megszokott, azt kódotlani kell, általában %xx formában, ahol xx a karakter hexadecimális kódja. (Még a szóközt is helyettesítik, +jellel). Ha URL-eket „építünk” fel az ASP oldalainkban, mindig használjuk ezt a metódust is. Az *urlenc.asp* oldalon kipróbálható a kódolási művelet.

A [Server.MapPath\(\)](#) nagyon gyakran használt metódus. Arra való, hogy a virtuális könyvtár- és fájlnevetek (*pl. http://localhost/asp4/default.asp*) valós fájlnevekké alakítsa (*pl. C:\inetpub\wwwroot\asp4\default.asp*). A [mappath.asp](#) segítségével ezt is ki lehet próbálni. Ha a MapPath()-nek átadott virtuális név „/” vagy „.” karakterrel kezdődik, akkor a virtuális gyökérkönyvtárolt, ellenkező esetben pedig a hívás helyétől relatív fájlnevekkel dolgozik (*nem lesz ugyanaz az eredménye tehát a „default.asp” és a „./default.asp” kódolásának – kivéve ha éppen a gyökérkönyvtárban „allunk”).* Természetesen használhatók a „.” és „.” karakterek, azaz mozoghatunk a virtuális könyvtárak által alkotott térben (*a „.” az aktuális, a „.” pedig a virtuális szülőkönyvtárra mutat*). A [Server.MapPath\(\)](#) metódust nem használhatjuk a *global.asa* Application\_OnEnd eseményének kezelése közben, egyébként pedig legyünk óvatosak, mert a *global.asa*-ban meghívott MapPath() nem a *global.asa*, hanem a felhasználó által megnyitni kívánt fájl alapján dolgozik.

A [Server.ScriptTimeout](#) beállításával a scriptek időtűlépésének határát növelhetjük meg (*ez az érték nem lehet alacsonyabb, mint a tulajdonságlapon beállított alapértelmezés*), illetve kérdezhetjük le.

A [Server.Transfer\(\)](#) és a [Server.Execute\(\)](#) két, az IIS5-ben új szolgáltatás. A [Server.Transfer\(\)](#) metódus egyszerűen átadja a vezérlést az alkalmazásban található másik .asp kódnak. Minden átvett adat megmarad, többek között a Request objektum teljes tartalma is. A [Server.Transfer\(\)](#) kiválthatja a [Response.Redirect\(\)](#) használatát, mert ehhez nincs szükség a böngésző közreműködésére (és így felesleges hálózati forgalomra). Természetesen ez a megoldás csak akkor működik, ha webkiszolgálón belül szeretnénk átírányítani a felhasználót.

A [Server.Execute\(\)](#) szubrutinszerűen végrehajta az adott oldalt, majd a végrehajtás befejezése után visszatér az eredetihez (*mintha csak beáruztuk volna a kódot*). Ennek előnye az `<!-- include -->` kitételrel szemben az, hogy itt akár dinamikusan is generálhatjuk a futtatandó oldal nevét.

A [Server.GetLastError\(\)](#) visszaadja az utolsó ASP hibának adatait tartalmazó ASPError objektumot. Erről, és az ASP oldalak hibakezeléséről a következő számban lesz szó.

Végül, de egyáltalán nem utolsósorban: a [Server.CreateObject\(\)](#) metódus létrehoz egy adott objektumot. Ezt már nagyon sokat használtuk, anélkül, hogy tudtuk volna, pontosan mit jelent. Ha a létrehozott objektum tartalmaz OnStartPage metódust, azt is meghívja.



A `Server.CreateObject()` mellett objektumok létrehozására használhatnánk egyszerűen a `CreateObject()` hívást is. Ez utóbbi nem az IIS, hanem a scriptmotor szolgáltatása, ezért használata az ASP oldalakban – néhány kivételtől eltekintve – nem ajánlott.

Az objektum addig marad életben, amíg azt nem töröljük, illetve az ASP oldal végrehajtása véget nem ér. Ha az objektumot a Session vagy Application objektumba töltöttük, akkor az objektum élete csak az adott Session vagy Application objektum megszűnésekor ér véget.

Egy létrehozott objektumot úgy törölhetünk, ha a változóinak más értéket adunk (akár szöveget is, de elterjedt és szép megoldás a `Nothing`):

```
' Létrehozzuk:
Set oFSO = Server.CreateObject(
    "Scripting.FileSystemObject")
' Megszüntetjük:
Set oFSO = Nothing
```

### Naplózás az eseménynaplóba

Néhány résszel ezelőtt bemutatuk a `Response.AppendToLog()` metódust, amelynek segítségével írni lehet az IIS szöveges naplójába. Az ilyen bejegyzések feldolgozása viszonylag nehézkes, ráadásul – ha még emlékszünk – vannak olyan naplóformátumok is, amikor ez a módszer nem vezet eredményre. Szerencsére a Windows Scripting Host-nak köszönhetően a feladatot sokkal elegánsabban is megoldhatjuk. A Windows Scripting Host (és vele együtt természetesen a teljes WSH objektummodell) már az Option Pack 4-tyel bekerült (hetet) a Windows NT 4.0-ba, és azóta természetesen a Windows 2000 scriptprogramozásának lelkét képezi. A WSH objektummodelljét szerencsére ASP oldalakból is elérhetjük (egy későbbi alkalommal majd összefoglaljuk a lehetőségek teljes palettáját, aki addig is kíváncsi lenne, a WSH objektummodell leírását itt találja: [4]).

Visszatérve a témához, a WScript.Shell objektum `LogEvent()` metódusa lehetővé teszi, hogy belekönytároljunk a Windows NT/2000 eseménynaplójába. (Windows 9x esetén a bejegyzések a `Windows könyvtárban`, a `wsh.log` fájlban jönnek létre). A `LogEvent()` metódus három paramétert vár, amiből az első kettőt kötelező megadni, a harmadiknak pedig csak Windows NT/2000-n van értelme:

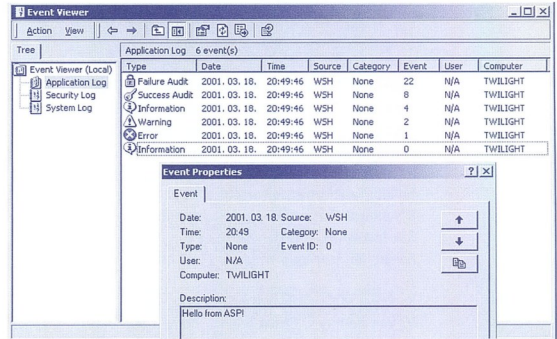
- ❶ A bejegyzés típusa (0: Success, 1: Error, 2: Warning, 4: Information, 8: Audit Success, 16: Audit Error)
- ❷ A bejegyzés szövege
- ❸ Opcionálisan a Windows NT/2000 számítógép neve, ahol a bejegyzést létre kell hozni (természetesen csak a megfelelő jogosultságok megléte esetén)

```
<%
Set oShell = Server.CreateObject("WScript.Shell")
```

```
oShell.LogEvent 0, "Hello from ASP!"
oShell.LogEvent 1, "ERROR from ASP!"
oShell.LogEvent 2, "Warning from ASP!"
oShell.LogEvent 4, "Info from ASP!"
oShell.LogEvent 8, "Audit Success from ASP!"
oShell.LogEvent 16, "Audit Error from ASP!"
```

```
Set oShell = Nothing
%>
```

A fenti példában (`logevent.asp`) a `Server.CreateObject()` metódus segítségével létrehozunk egy Shell objektumot, elkészítettünk néhány bejegyzést, majd jó kiscerkész módjára kitarítottunk magunk után (az objektumváltozó értékét `Nothing-ra` állítva felszabadítjuk az objektum által elfoglalt erőforrásokat). Az eredmény az ábrán látható:



☞ A `WScript.Shell` objektum segítségével készített bejegyzések az `Application Log` naplóban láthatók

A generált naplóbejegyzések (az `audit log` is) az eseménynapló `Application Log`-jába kerülnek, a `Source` mezőben a „WSH” felirat szerepel. Ha a bejegyzést megnyitjuk, a leírás mezőben láthatjuk az általunk megadott szöveget.

Fülöp Miklós  
mick@netacademia.net

### A cikkben található URL-ek:

- [1] <http://technet.netacademia.net/feladatok/asp/4>
- [2] <http://msdn.microsoft.com/library/psdk/iisref/crtc796b.htm>
- [3] <http://www.microsoft.com/technet/security/crssite.asp>
- [4] <http://msdn.microsoft.com/scripting/windowshost/doc/wshstoc.htm>

### ASP objektummodell-referencia a weben:

<http://msdn.microsoft.com/library/psdk/iisref/vb074bw.htm>



**K:** Megpróbáltam a kiszolgálón átállítani a fájltypus-összerendeléseket, illetve megpróbáltam volna, ha nem lenne szürke a menüpont. (Explorerben Tools->Folder options->File types). Úgy érzem, nem követtem el semmi olyasmit, amivel ezt a bűntetést kiérdemeltem. Miért szürke?

**V:** Van egy-két ilyen eltűnedező szolgáltatás, egy korábbi kérdésben például arra panaszkodott valaki, hogy az Offline Files tünt el szűrőstül-bőrőstül. Mindkét korlátozás a Terminal Services telepítése miatt következett be. A Terminal Server futtatása összeegyeztethetetlen bizonyos más funkciók használatával. Sajnos nem tudjuk megmagyarázni, hogy a fájlkiterjesztés megváltoztatása, és az Offline Files mit árthat a Terminal Servernek – ezt Redmondban biztosan meg tudnánk indokolni – mindenestre ez van, tehát ezt kell szeretni. *Forrás: NetAcademia Windows 2000 lista*

**K:** Van-e DOS-os kliens Windows 2000-hez?

**V:** Van de-nincs. Ez azt jelenti, hogy a régi DOS-os kliensek természetesen tökéletesen működnek a Windows 2000-re, de eltűnt az operációs rendszerből az NT4-ből megismert flopigenerátor. Azaz, ha nincsen készen ilyen DOS-os kliens, akkor egy jó öreg NT4 segítségével generálhatjuk le a hálózati kapcsolatra alkalmas floppyt. A Windows 2000-ben található Remoteboot Floppy Generator (RBF.GEN) nem erre való, az ugyanis a Remote Installation Serviceshez (RIS) gyárt indítólemezt – de az nem dosos, hanem PXE (Preboot Execution Environment) „operációs rendszerű”, és semmi másra nem alkalmas, mint Windows 2000 Professional telepítésére. *Forrás: NetAcademia Windows 2000 lista*

**K:** A Group Policy működésével kapcsolatban merült fel az igény, hogy ne kelljen kivágni a munkaadóknál a 90 perces késletetést a kiértékelés automatikus beindulásáig, de ne kelljen ki-be jelentkezni, pláne újraindítani a gépet. Nem lehetne valahogy „megsürgetni” a GPO-t?

**V:** A Group Policy kiértékelése szándékosan késletetett, hisz ezzel jelentős sávszélesség takarítható meg. Ha mégis szükség van a változtatások gyors, menet közbeni kiértékelésére, használjuk a SECEDIT.EXE-t parancssorból. Mivel a GPO számítógépekre és felhasználókra is hathat, kétféle parancs van ezek külön-külön frissítésére. Ha gépszintű beállítást szeretnénk hirtelen frissíteni, (például a Log On Locally jog megváltozását szeretnénk életbe léptetni) használjuk a

```
SECEDIT /REFRESHPOLICY MACHINE_POLICY
```

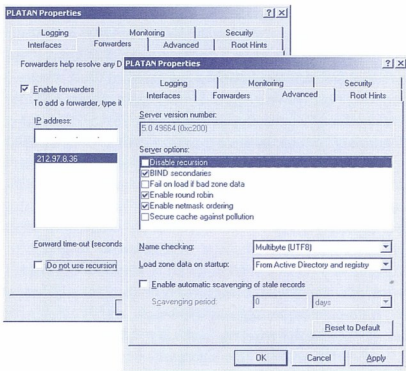
parancsot, míg ha felhasználói beállításról (tűnjön el a RUN menüpont!) van szó, a parancs így néz ki:

```
SECEDIT /REFRESHPOLICY USER_POLICY
```

*Forrás: NetAcademia Windows 2000 lista*

**K:** A DNS Server tulajdonosglapján két helyen is van „recursion” pipa. Feltételezem, hogy tudom mi a DNS rekurzió, mi a különbség a két pipa között?

**V:** A különbség óriási. A Forwarders fülön lévő pipa hatására a DNS Serverünk a helyben megválaszolhatatlan kéréseket csak és kizárólag a beállított forwarderek-nek küldi tovább, s ha azok nem válaszolnak, hibábaüzenetet küld a kérdezőnek. Azaz ez a pipa „csak” annyiban érinti a munkaállomásokot, hogy ha a forwarder DNS kiszolgáló lehal, akkor „vége a külföldnek”, egyéb változás nincs. Ezzel szemben az Advanced fülön lévő beállítás súlyosan visszahat az ügyfélgépekre, mert azt mondja ki, hogy a DNS Server ne járjon el (rekurzíve) a neten a kliensek nevében, hanem ha egy kérdést nem sikerül helyben feloldania, akkor a továbblépés útjára mutató ügyvezetett Referral rekordot küldjön a kliensnek – oldja meg a problémát maga! No ez utóbbi tipikusan nem fog menni a munkaállomás belsejéről. Ezt a pipát NE bántsd!



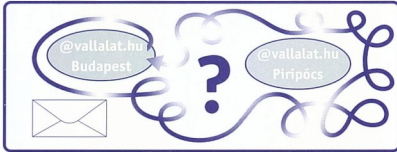
☞ **A két Recursion pipa**

*Forrás: NetAcademia Windows 2000 lista*

**K:** Van két Exchange 5.5 telephelyünk, melyek közül eddig csak az egyikben volt SMTP konnector. Most a másik telephely is kapott bérlet vonalat, és ugyanazt az SMTP email címet szeretnénk használni. Be is állítottam mindkét telephelyen a Site Addressingben a @vallalat.hu címet, – erre a két telephely leszakadt egymásról, megszűnt a belső levelezésünk!

**V:** A probléma oka természetesen az azonos SMTP cím, mert így a szomszéd telephelyre menő leveleket az Exchange nem küldi át, hisz a helybéli telephely „közelebb van”. Magyarul belezavarodik.

@vallalat.hu



☞ *A levél útja két azonos nevű SMTP tartomány között :)*

Ám itt meg – értelemszerűen – nincs olyan nevű mailbox, mint ahova a levél menne. Így a levelezésnek valóban annyi. Megoldási lehetőségek:

- 1) Ne legyen azonos a két telephely SMTP címe. Ez hosszas folyamat, DNS regisztráció, MX rekord, névjegykártyacsere az összes dolgozónál stb.
- 2) Ha a cím marad, akkor pedig NE legyen két SMTP konnektorkunk. A másik telephelyen le kell szedni! S hogy akkor a telephelyek hogyan levelezzenek egymással az Interneten? X.400-zal!
- 3) Térjünk át Exchange 2000-re. Ott ez a problémát kezelzi a sokkal intelligensebb routing algoritmus.

*Forrás: NetAcademia Exchange 2000 lista*

**K:** *Adott egy házi rend ami egy szervezeti egységhez tartozik. Ezt a házi rendet szeretném alkalmazni kisebb módosításokkal egy másik szervezeti egységénél is. Hogyan lehet megoldani azt, hogy ne kelljen újra beállítani minden egyes szervezeti egységénél ugyanazt, hanem újra fel lehessen használni az egyszer már megírt házi rendet?*

**V:** Úgy, hogy a következő felhasználási helyen (OU-n) nem New.. hanem Add.. gombot nyomsz, és ezzel egyszerűen belinkel a már meglévő policy-t. Ám ekkor egyetlenegy példányod lesz, tehát ha az egyik helyen módosítod, a másikon is módosul. Ha valóban másik, ám ugyanolyan policy-t kellene létrehozni, akkor a következő módszer segít:

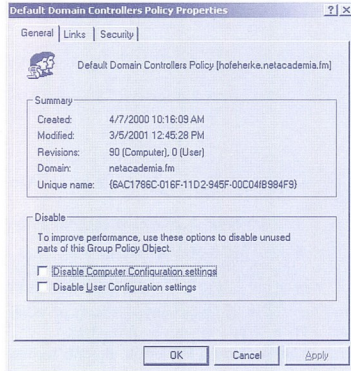
- 1) Áss le a policy-fájlokhoz a `winnt\sysvol\sysvol\domained\policies` könyvtárba. Itt ronda, GUID nevű alkönyvtárakat látsz.



☞ *GUID nevű könyvtárak*

Ebben ülnek a policy-fájlok. Most már csak meg kellene határozni, hogy melyik melyik.

- 2) Nyisd meg az OU-t az Active Directoryban, állj rá a polycyra, és nyomj „Properties”-t. Itt nézd meg a policy GUID-jét: az egyik könyvtárnévvel egyeznie kell.



☞ *A policy GUID-je*

Innentől már csak kipi-kopi kérdése az egész!

*Forrás: NetAcademia Windows 2000 lista*

**K:** *Hogyan lehetne megszabadulni az Accessoires, Games, Accessibility és egyéb haszontalanságtól Windows 2000 esetén? Sajnos az Add/Remove Windows Components nem teszi lehetővé az eltávolításukat.*

**V:** A Windows 2000 Professional telepítésekor a Games, Accessories, Multimedia, Accessibility options valóban automatikusan települ, és ezután már a Control Panel Add/Remove Programs menüjében sem jelenik meg. De van megoldás:

- 1) Keressük meg, és nyissuk meg Notepad-ben a `%systemroot%\inf\sysoc.inf` fájlt
- 2) Keressük meg az "old base components" sort
- 3) Minden megjeleníteni kívánt komponensnél töröljük a HIDE feliratot (*hogy a vesszőt is kell-e, az nem egyértelmű*), pl.:  
`Games=ocgen.dll,0cEntry,games.inf,HIDE,7` helyett  
`Games=ocgen.dll,0cEntry,games.inf,,7`
- 4) Töröljük a HIDE opciót az AccessUtil= sorból
- 5) Save and Enjoy

*Forrás: NetAcademia Windows 2000 lista*



# Bárki Bármikor Bárhol

## A következő technológiai lépés a mindenre használható hozzáférés lesz.

Meglepő, de a „konvergencia” fogalma már több mint húsz éve bekerült a csúcstechnika lexikonjába.

Az elmúlt évek nagyobb részében a konvergencia két dolgot jelentett: a számítástechnikai, a szórakoztató elektronikai és a telekommunikációs iparágak közeledését, és az eszközök, például a PC, a TV és a telefon összevonását.

De a 21. század közeledtével a kifinomult digitális technológiák és a sávszélesség robbanásszerű növekedésének ígérete egy harmadik féle konvergenciát hoz létre, mely alapvetően megváltoztatja az életünket, jobban, mint ahogy azt eddig gondoltuk volna. Ez fogja biztosítani az információ korának előnyeit bárkinek, bármikor, bárhol.

Olyan korban élünk, melyben a hang, az adatok és a mozgóképek bitekből állnak. Egyesekből és nullákból, melyeket a lehető legnagyobb sávszélességű, vagy az adott szolgáltatáshoz legalkalmasabb adatátviteli csatornán juttatnak el hozzánk. A bitenként nem számít, hogyan jutnak el rendeltetési helyükre, csak az, hogy a megfelelő sorrendben és időben érjenek oda. Az, hogy a bitek mindenütt jelen vannak, máris lehetőséget biztosít olyan hibrid készülékek (például a modern PC-k, WebTV-k, kábelmodemek, intelligens telefonok) létrehozására, melyeket az utóbbi két évtizedben elektronikai termékként megálmodtunk.

De ez még csak a kezdet. Ötvözzük a digitális technológiát és a fejlett szoftvereket, a kisebb és erősebb mikroprocesszorokat, az üvegszálas és vezeték nélküli hálózatok sávszélességének exponenciális növekedését, és sokkal jobban felhasználható, észrevétlen, mindenre alkalmas kapcsolatokat kapunk. Ez megváltoztatja a konvergenciát, azaz, bár a számítógépek, a telekommunikáció és a szórakoztató elektronikai termékek összevonódnak, az intelligens eszközök következő generációja általában nem.

Ezek a mindenre használható kapcsolatok összegyűjtik a számunkra szükséges információkat és szolgáltatásokat, és elérhetővé teszik azokat, függetlenül attól, hogy hol vagyunk, mit csinálunk, és milyen eszközt használunk. Nevezük ezt virtuális konvergenciának, hiszen minden, amit akarunk egy helyen van, de ez a hely mindig ott van, ahol akarjuk, nem csak otthon, vagy az irodában.

Ennek eredménye olyan intelligens eszközök halmaza lesz, (például a zseb PC-k, a „tábla PC-k”, a webes képességekkel rendelkező telefonok és autóbá szerelt PC-k), melyek kihasználják ezeket a lehetőségeket. A fájljaink, az időbeosztásunk, a címjegyzékünk és bármi más amire szükségünk lehet, automatikusan replikálódni fog ezekre az eszközökre, mert mindegyik össze tud majd kapcsolódni egymással.

A virtuális konvergencia átveszi az asztali gépeink szerepét, és mindenhol elérhetővé teszi az információkat, ahol csak szükségünk van azokra. De hogy segítsen elkerülni azt, hogy túl sok információhoz jussunk, intelligensen fogja ezt végrehajta-

ni. Ha például szombat délután meccset nézünk a stadionban, a hálózat tudni fogja, hogy a játékosokról szóló információk, vagy például egy online hotdog-rendelési űrlap zseb PC-nként való megjelenítése szükséges, és nem az adóbevallásunk, vagy a másnapi utazásunk útítévére vagyunk kíváncsiak.

Még sok mindent kell tenni, hogy ez valóra válhasson. Elsősorban nagyjából ugyanazt, mint amikor az emberek azonos nyelven beszélnek. Ahhoz, hogy megértsek egymást, hatékonyan együttműködjenek és kommunikáljanak egymással, az intelligens eszközöknek is közös nyelvet kell „beszélniük”. Ennek legegyszerűbb és legcélszerűbb módja a létező Internetes szabványok alkalmazása. Az iparág erre felé mozdul, de még mindig igen sok a tennivaló. Ez azonban még nem elég, jelentős tőkebefektetésre is szükség van, ha azt akarjuk, hogy a nagysebességű, szélessávú kommunikáció széles körben elérhető legyen az Egyesült Államokban (*Hát még Magyarországon... (- A ford.)*).

Hogy ez megvalósulhasson, egyre több megállapodás, szövetség és konzorcium jön létre, melyekben számítástechnikai, szórakoztató elektronikai, telekommunikációs, Internetes és kábelhálózatokat üzemeltető cégeknek van érdekeltségük. Mivel mindenkinek szembe kell nézni az életképes üzleti modell kialakításának kihívásával, e cégek némelyike tönkre fog menni. Két dolog azonban biztos: a mindenhol elérhető vezeték nélküli, és a nagy sávszélességű adathálózatok kezdenek kiépülni, és a különféle intelligens eszközök, melyekre ezeknek a hálózatoknak az eléréséhez szükség lesz, hamarosan megjelennek a piacon. Ezek együttese valóra válthatja a virtuális konvergenciában rejlő lehetőségeket.

Bill Gates



## TechNet szemináriumok a Microsoft Magyarország szervezésében!

Helyszín: BUDAPEST, Lurdy Ház, Hollywood Multiplex Mozi (1097 Budapest, Könyves Kálmán krt. 12-14.)

### Microsoft Üzemeltetői Konferencia

**Időpont: 2001. 04. 25.**

A konferencia elsősorban üzemeltetési vezetőknek és rendszerüzemeltetőknek szól. Segítséget nyújt a nagy- és középvallalatok informatikusainak és rendszergazdáinak, hogy a már meglévő Microsoft technológiákra épülő rendszereiket még megbízhatóbban és egyszerűbben üzemeltethessék. Szeretnénk bemutatni és átadni mindazt a tudást, ami a Microsoftnál és partnereinél az üzemeltetési és terméktámogatási gyakorlat során felhalmozódott.

### Rendszerfelügyelet Windows 2000 környezetben

**Időpont: 2001. 05. 02.**

Az előadás a Windows 2000 alapú vállalati környezetek rendszerfelügyeleti és üzemeltetési kérdéseit tárgyalja. A gyakorlatias megközelítésű előadáson egy kiépített Windows 2000 Active Directory, Systems Management Server (SMS) 2.0 és Microsoft Operations Manager környezetben mutatjuk be az asztali gépek és alkalmazások, illetve a kiszolgálók felügyeletét és a napi üzemeltetési teendőket.

### Üzleti Intelligencia: adatraktárak létrehozása és az adatok elemzése

**Időpont: 2001. 05. 16.**

Minden vállalat használ adatbáziskezelőt, de vajon kihasználja-e a felhalmozott üzleti célú adatokban rejlő információkat? Az előadások az adattárház kezelés folyamatának legfontosabb lépéseit tekintik át. Ismertett technológiák: Data Transformation Services, OLAP kiszolgáló (*Analysis Services*), MDX, ADO DM.

Rendszerórákat szinkronizálni a Windows 2000 alatt?  
Minden alhálózatra globalis katalógust tenni?  
Certificate Server-t telepíteni?  
Sokan megtudják mondani, hogy HOGYAN...  
De tőlünk azt is megtudja,  
hogy MIÉRT!  
[www.netacademia.net](http://www.netacademia.net)

NETACADEMIA  
A legjobbakat keressük!

# Halott már Ön selejtmentes gyártásról? Rendszer- felügyeletmentes hálózatról (ZAK, ZAW)? Feltörhetően hálózatról (C2)?

Elérhetetlen, de megközelíthető célok. Az első kettő elérésében sajnos nem segíthetünk. De a harmadik cél megközelítésében sokat tehetünk Önért. Világszínvonalú, egyedi Security tanfolyamainkon megtudhatja, mitől döglök a hacker!

## 1. Biztonsági technológiák rendszergazdáknak, 3 nap

- Titkosítási algoritmusok (DES, RSA, MD5 stb.) elmélete és gyakorlata
- Az autentikáció biztonsága (Kerberos, NTLM, NTLMv2)
- A nyílt kulcsú titkosítási technológia nagyvállalati, gyakorlati alkalmazása (Certificate Server, SmartCard, biometrikus azonosítás)
- Biztonságot növelő technológiák bevezetése (IPSec, VPN, L2TP, Encrypting File System stb.)

## 2. Biztonságos levelezőrendszer kialakítása MS Exchange alapokon gyakorlott Exchange rendszergazdák számára, 3 nap

- Titkosítás és digitális aláírás az elektronikus levelezésben (S/MIME)
- Az Exchange Server védelme, adminisztratív szerepek, jogosultságok, mentés, visszaállítás, journaling
- Tűzfal vs. Exchange Server, SMTP relay beállítások, spam védelem, SSL, TSL, nyílt kulcsú titkosítás felhasználása, MAPI, POP3, IMAP, OWA, LDAP protokollok veszélyei.

## 3. Hálózatunk védelme a támadásokkal szemben biztonsággal foglalkozó szakembereknek, 3 nap

- Kalóz-eszközökészlet (Denial Of Service, Buffer Overrun, hálózatelemzés, trójai falovak, vírusok, jelszófeltörés, hátsó ajtó)
- A Windows 2000 biztonsági elemei (fájlrendszerek, regisztrációs adatbázis, hálózati adatforgalom)
- Védekezés: tűzfalak, proxyk
- Aktív védekezés: Intrusion Detection

## 4. Programozunk biztonságosan! programozók számára, 3 nap

- Elmélet: védett mód, kernel, memóriakezelés
- Programozási gyakorlatok, rosszul és jól megírt kódok
- A Buffer Overrun hatásai user és (app es service)+kernel módban
- RFC implementation errors
- JAVA, ActiveX, homokozó, scriptek
- Worm vírusok elemzése (loveyou)
- CryptoAPI, Smart Card API

Biztonsága érdekében keressen minket!

[www.netacademia.net](http://www.netacademia.net)

A legjobbakat tanítjuk!

1105 Budapest, Ilhász utca 13. • Tel.:263-2732

