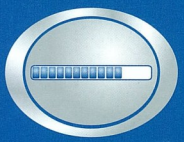
tech.net
working with windows
III./07.szám
1344 Ft

Windows 2000 IP Security

13. oldal



ISSN 15865185
07
9 771586 518005

A LEGENDÁS ÖT KILENCES. KÖZELEBB VAN MINT GONDOLNÁ!

A szerver operációs rendszereknél, az a bizonyos öt kilences a megbízhatóság legfelsőbb fokát jelzi. Gyakorlati nyelvre lefordítva, ez a teljes rendelkezésre állás mellett, mindössze 5 perc kiesést jelent évente!* A 99,999%-os megbízhatósági mutató már nem csak laboratóriumi körülmények között érhető el. Már nem csak olvasni lehet róla. Már nem csak a méregdrága számítástechnikai rendszerek kiváltsága. Közelebb van mint gondolná! Képzelje el, hogy már az Ön cége számára is elérhető ez a szédületesen stabil szerver háttér! Képzelje el, hogy mindez kedvező áron megvásárolható! Képzelje el, hogy nem kell többé elképzelnie, mert a 99,999%-os megbízhatóság az Ön cége számára is valósággá válhat!

Ha a **Microsoft Windows 2000 Server** családot választja, akkor a megbízhatóság csúcsát választja.

Microsoft Windows 2000 Server Family

*Ez az adat függ az operációs rendszeren kívüli körülményektől is, mint például a hardvereszközök milyenségétől, illetve más, az operációs rendszertől független szoftverektől.

99999 A legendás öt kilences: 99,999%-os rendelkezésre állás.

http://www.microsoft.com/hun/products/windows.htm
http://www.microsoft.com/hun

Microsoft Ügyfélszolgálat: 2MSINFO [267-4636]
Microsoft Forródrót: 2MSSUGO [267-7846]

Microsoft

Glokalizáció

tech.net
working with windows

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@technetklub.hu

Kiadja és terjeszti a
NetACADEMIA
A LEGJOBBAKAT TANÍTJUK

NetAcademia Kft.
Terjesztési, előfizetési információ:
Tel.: 263-2732
terjesztes@netacademia.net
Megjelenik havonta, ára 1.344 Ft
Példányszám: 4.000

Előfizethető megrendelőlevélben a szerkesztőségnél:
1105 Budapest, Ihász utca 13.
Fax: 261-7145
http://technet.netacademia.net/subs

Hirdetésfelvétel:
BÁRSONYKALAPÁCS
MARKETING
VELVET HAMMER MARKETING
Az ütütt siker kovácsa.
Felelős: **Balogh Zoltán**
Tel.: 489-4661
Fax: 489-4660
info@velvethammer.hu
1027 Budapest, Fő utca 67. V. 1.
Grafikai tervezés, kivitelezés, nyomdai előkészítés:
Bársonykalapács Marketing
Művészeti vezető: **Balogh Zoltán**
Bársonykalapács © Copyright 2002

Nyomda:
Hieron Kft.
2120 Dunakeszi, Tamási A. u. 11/a.
Felelős vezető: **Török Andrea**

ISSN 1586-5185

Elmúlt számunkban megígértem, hogy beszámolok a barcelonai tech.edről. Sajnos ezt igen röviden meg tudom tenni: vegyék elő tavaly augusztusi számunkat, mert abban minden benne van. Barcelona sem sokat változott, az **[1]** címen tavaly óta megtalálható élménybeszámoló tehát teljesen aktuális. Volt egy csomó híres előadó, de szinte mindegyikük a tavalyi PPT-je fölött unatkozott. Hogy ennek mi lehetett az oka, nem tudom, de annyiban szándékosnak tekinthető, hogy a konferencia jelmondata is ismétlésre utalt: dive deeper. Hirtelen fordításban: búvárkodj a mélyben. Helyes fordításban: ismétlés a tudás anyja.

Így az előadásokról nem, de benyomásaimról, és egy heti önálló gondolkodásomról beszámolok. Ez a rendhagyó köszöntő a 43-44. oldalon folytatódik. Ha azt hiszik, a hőség ment az agyamra, közlöm, nem a hőség. Légkondicionált helyiségben írtam e sorokat.

Nagy örömömre szolgált, hogy Don Box, a SOAP atyja hasonló okfejtést adott elő - igaz, a webszolgáltatások témakörében. Milyen kihívásokkal szembesülnek napjaink vállalatai? Hisz már mindenen túlvannak, övék az Active Directory! Igen ám, de odalett az autonómia...

Vegyünk egy tipikus céget, mely az elmúlt években kitartóan törekedett a vállalatban elszórtan megtalálható információmorzsák központosítására. A clipperes alkalmazásokat kihajították, és helyükre vadonatúj, ügyfél-kiszolgáló felépítésű, valódi adatbázismotoros alkalmazás került. A hálózati szigeteket bérelt vonallal összekapcsolták, és véres verejtékkel tartománykonszolidációt hajtottak végre, hogy a négyszáz tartományt és munkacsoportot egy Active Directoryra cseréljék le.

A nyereség könnyen kimutatható, s a korábbi ígéreteknek megfelelő: a cég a központi adatbázissal okosabb, a központosított rendszerfelügyelettel rugalmasabb lett, s könnyebben él együtt azzal a helyzettel, hogy távoli telephelyeken nincs „értelmes ember", akire feladatokat lehetne bízni.

Ez mind igaz. Nemrégiben mégis csúnya konfliktusba keveredtem egy hazai bankban, ahol egy elszigetelten működő, több szerverből, valamint egy saját NT4 tartományból álló kritikus alkalmazás rendszergazdája kézzel-lábbal tiltakozott az Active Directoryba belépés, s ezzel egy elfogadott fejlesztési terv ellen. Kézenfekvőnek tűnik, hogy Active Directory bevezetésekor minden korábbi tartományt és meghatalmazotti viszonyt *(trust)* oldjunk fel, és irtsuk ki a központi felügyeletet gátló tartományocskákat. Közelebbről megvizsgálva az alkalmazásszigetet, az derült ki, hogy van ugyan egy saját tartományvezérlője, de csak azért, mert az SQL Server fürtön fut, a Cluster Service pedig tartományi fiókot igényel. Az SQL Servernek saját felhasználója nincs, egy nagygéppel textfájlokon keresztül kommunikál, és kész. Akár hiszik, akár nem, egyetlen felhasználó kallódott abban a tartományban *(és erre sem lett volna szükség, ha a fürt nem olyan, amilyen)*. A rendszer évek óta változatlan formában fut, és még egyszer sem állt le. Mármost milyen előnyök származhatnának e tartomány beolvasztásából? Könnyebb rendszerfelügyelet? Aligha! Egyik legnagyobb erőssége, hogy hónapszámra hozzá sem kell nyúlni! Ha előny nincs, hátrány származhat-e az „előrelépésből"?

És itt veszett össze az Active Directory megtervezésével és bevezetésével megbízott cég a belső rendszergazdával, a konfliktus feloldására külső szakértőként engem kértek fel. A problémát lecsupaszítva, politikai, érzelmi és egyéb vetületektől megtisztítva ez marad: biztosítható-e, hogy egy *(többezer felhasználós)* Active Directoryban ne történhessen olyan változás, amely padlóra küldheti az eddig önállóan és zökkenőmentesen futó alkalmazást? Ha készítenénk egy nyilatkozatot erről, vajon lenne-e valaki, aki ezt aláírná?

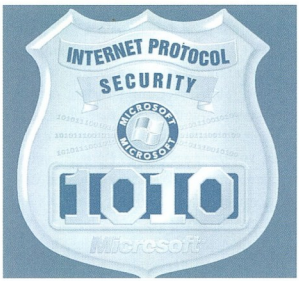
Két hónappal ezelőtt csupán a józan paraszti eszemre hallgatva utasítottam el, hogy személyes garanciát vállaljak egy tőlem független rendszer viselkedéséért. A sors azonban úgy hozta, hogy pontosan a fenti dilemmára esettanulmány született egy másik cégnél. Lepenye Tamás már készíti a következő „Ki mivel ...hmmm" cikket, melyben elolvashatjuk, mint borította fel egyébként tökéletes fürtjét egy pár hónappal korábbi AD-módosítás. Ha még az ő keze között is felborul, aki nálam tízszer alaposabban készít elő mindent, a változtatások előtt hősiesen elolvassa az összes vonatkozó Knowledge Base cikket, ki kell mondanunk: ez mindenkivel előfordulhat.

Központi hibaforrás?

Tudomásul kell vennünk, hogy központosított rendszereinkben a hibaforrás is központosított, a hiba hatása pedig globális. Vegyünk egy igen egyszerű példát: kézi, vagy automatikus IP-cím kiosztást válasszunk? 500 gép esetén nem kérdés, hogy az utóbbit, DHCP-s megoldást választunk, mert nincs olyan szupermen ebben az országban, aki akárcsak megközelítőleg olyan gyorsan és pontosan kiosztaná az IP-címeket, mint egy direkt erre a célra kitalált címosztó automata. Megszüntettünk ötszáz pici hibalehetőséget - helyette kaptunk egy nagyot. Ha egy felelőtlen

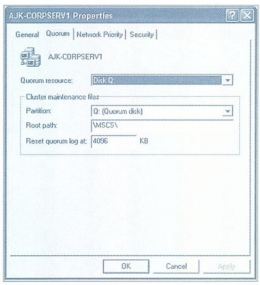
Folytatás a 43-44. oldalon

2002 Július



Windows 2000 IP Security

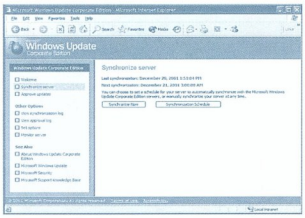
13. oldal



Farkasokkal táncoló

Az előző alkalommal a hibabehatárolás és hibaelhárítás tudományába kóstoltunk bele. Ezúttal olyan helyzeteket próbálunk megoldani, amikor a fürtöt olyan alapvető behatás éri, ami nem feltétlenül hiba, de akár az is lehet. Olyasmi szituációkat elemzünk, amikor „a Föld fog sarkából kidőlni...”

4. oldal



Windows Update Corporate Edition

Ez a cikk a Windows Update Corporate Editionről *(WUCE)* szól. Ez az eszköz megkönnyíti az Internetről letölthető Windows javítások vállalaton belüli telepítését. A WUCE a jól ismert Windows Update szolgáltatás továbbfejlesztése, segítségével megoldható, hogy a vállalati számítógépek ne közvetlenül az Internetről, hanem belső, vállalati kiszolgálókról töltsék le a javításokat.

7. oldal

Digital Encryption Standard

Ritkán megjelenő kriptográfia-sorozatomban a tavaly decemberi nyílt kulcsú titkosítás *(RSA-algoritmus)* után most a szimmetrikus titkosítások legelterjedtebb képviselőjét, a DES *(Digital Encryption Standard)* eljárást nézzük meg értő szemmel. Mire a cikk végére érünk, mindenki tudni fogja, miként működik egy szubsztitúciós-permutációs hálót használó, CBC módú Feistel-cipher.

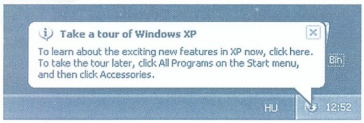
17. oldal

P			
16	7	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	24	14
32	27	3	9
19	13	30	6
22	11	4	25

XP: tömörítések

A Windows XP összes változata *(Professional, Home)* NT alapú. Ha pedig NT, értelemszerűen a korszerű, modern NTFS fájlrendszert választjuk, nem a régi öreg, korlátolt FAT-et, vagy tuningolt változatát, a FAT32-t. Ezzel kismillió szolgáltatáshoz *(tranzakciónapló, jogosultsági rendszer, kvóta stb.)*, jutunk, és kezünkbe kerül az on-the fly *(röptében)* tömörítés is. Így már két tömörítési eljárás közül választhatunk, hisz az XP „zipelni” is tud. Mikor melyiket érdemes használni?

22. oldal



A Windows XP zavaró újdonságai

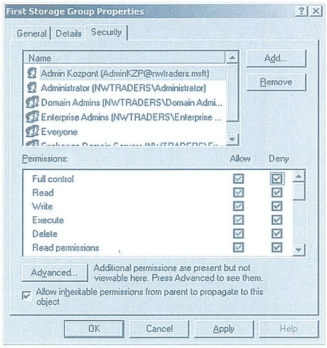
25. oldal

Microsoft Exchange 2000

Rendszergazdai jogosultságok

Csakúgy, mint az Active Directoryban, az Exchange-ben is van lehetőség arra, hogy meghatározott jogosultságokkal rendelkező kisrendszergazdákat nevezzünk ki. Ennek legegyszerűbb módja, ha a System Managerben az Exchange Administration Delegation Wizardot, vagyis a jogosultságosztó varázslót használjuk.

29. oldal



.NET Akadémia

Tömbök és kollekciók rendezése, keresés, hashelés

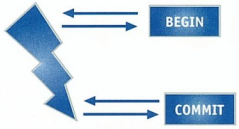
Az előző részben áttekintettük a kollekciók legfontosabb jellemzőit, megbeszéltük a hatékonysági kérdéseket, és láttuk, hogy a tömbök sokkal okosabbak mint azt elsőre feltételeznénk róluk. Láttuk az ArrayList osztály használatát, és elkezdtünk egy formos keresési példát. Ebben a részben megbeszéljük annak részleteit, és áttekintjük a maradék kollekcióosztályok működését.

33. oldal

Az Exchange 2000 Server és Web Storage System: *Event Sinkek*

Véleményem szerint az egyik legizgalmasabb dolog az informatikában az, amikor a dolgok „maguktól történnek". Levelek jönnek-mennek, dokumentumok megváltoznak, mintha valami eleven kobold ülne a gépünk belsejében. A kobold megszelídíthető, sőt engedelmességre is kényszeríthető, csak meg kell tanulnunk, hogyan szóljunk hozzá. A jelszó: Event Sink!

37. oldal



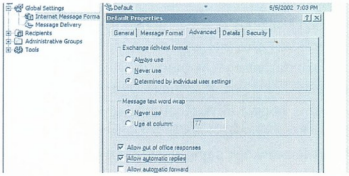
Patchwork

Javítások Excelhez Wordhöz

40. oldal

Dupla KV

41. oldal



Farkasokkal táncoló

(IX. rész) – Cluster a gyakorlatban

Az előző alkalommal a hibabehatárolás és hibaelhárítás tudományába kóstoltunk bele. Ezúttal olyan helyzeteket próbálunk megoldani, amikor a fürtöt olyan alapvető behatás éri, ami nem feltétlenül hiba, de akár az is lehet. Olyasmi szituációkat elemzünk, amikor „a Föld fog sarkából kidőlni...”

Lemezcsere – földrengés Kaliforniában

Többször vadásztunk már olyan pontokra, amelyek meghibásodása a teljes fürtöt használhatatlanná teszi. Találtunk is ilyeneket, és ha tudtuk, megszüntettük őket. Vannak azonban olyan összetevők, amelyekből nem lehet több példányt használni – ezek a lemezek. Persze replikázhat az Olvasó, hogy azokat lemeztömbökbe lehet szervezni, így biztosítható az állandóságuk. Nem tudok teljesen egyetérteni. Az állandóság biztosításáról nem érdemes beszélni, ha például tűzvész pusztítja el a tömböt, de ennél kevésbé tragikus események hatására is eltűnhetnek. Egy lemez ugyanis összetett dolog: nemcsak a lemezterületből áll, hanem az őt azonosító betűjelből is, sőt egy lemeznek szignatúrája, vagyis belső azonosítója is van a Windows NT – Windows 2000 világban. Ha bármelyik változik, a fürtszolgáltatásunk bizony azt gondolja, hogy a régi lemez eltűnt, és van itt valami új, amiről fogalma sincs, hogyan került ide. Erről úgy tájékoztat minket, hogy az adott lemezerőforrás „hibás” állapotba kerül ***(quorum disk esetén a fürt el sem indul)***, az eseménynaplóban pedig az alábbi bejegyzést találjuk:

```
Event ID: 1034
   Source: ClusDisk
   Description: The disk associated with cluster
disk resource <DriveLetter> could not be found.
The expected signature of the disk was
<DiskSignature>.
```

Magyarán az adott betűjellel jelölt lemezt a fürt nem találta. Azt a lemezt egy <DiskSignature> kód azonosította volna.
A hiba beszédes, és messzemenő következtetéseket lehet levonni belőle.

1. A fürt a lemezerőforrásokat a betűjelük alapján tárolja, a fizikai lemezekhez pedig a lemezek szignatúrájának segítségével kapcsolja. Vagyis: egy lemez mindaddig „önmaga”, amíg szignatúrája és/vagy betűjele nem változik.
2. A lemezek szignatúrája nem ismeretlen fogalom. A Windows NT a dinamikus lemezekig bezárólag ezt a módszert alkalmazta a diszkek azonosítására. Tulajdonképpen egy regisztrációs kulcsról van szó, ami egyúttal a rendszerről rendszerre való hordozhatatlanságát is mutatja. Megérdemelten váltotta le a módszert a dinamikus lemez. Mindebből azonban az is következik, hogy a fürtszolgáltatás egy kicsit elmaradt a korától és környezetétől. Mivel a szignatúrát használja, a dinamikus lemezeknek pedig ilyen nincs, ezért a fürt és a dinamikus lemez nem fér meg egymással.
3. A szignatúrák ismeretével és használatával könnyedén túljárhatunk a fürtszolgáltatás eszén. Ha viszont a lemezek ezen fontos adatai elvesznek, nagy bajban vagyunk!

Lássuk tehát, hogy a megszerzett ismeretekkel hogyan lehet szakszerűen elvégezni a lemezek cseréjét. Legyen életszagú a példa: egy lemezterületet kinőtt egy alkalmazás. Az alatta lévő hardver lemeztömböt már sikerült bővíteni, most azt kell elérnünk, hogy az „L:” meghajtó által meghatározott lemez mérete megnövekedjék. A feladatlista a következő:

1. Teljes mentés minden meghajtóról.
2. A fürt leállítása.
3. A lemez szignatúrájának megjegyzése.
4. A lemez törlése a RAID alrendszer szintjén.
5. Egy új, már megnövelt területű lemez definiálása.
6. A lemez formázása és betűjellel való ellátása.
7. A lemez szignatúrájának megváltoztatása az eredeti jelsorra.
8. Adatvisszatöltés.
9. A fürt indítása.

A teljes mentés világos feladat, minden beavatkozás első lépése. Cluster esetén ez annyit jelent, hogy a mappák és állományok mentése mellett a rendszerállapotot *(system state)* is menteni kell. A fürt leállítása már egy kicsit összetettebb. A menete a következő:

1. Állítsuk át az egyik állomáson a clustdisk és a cluster szolgáltatás indulását „kézi” vagy „tiltott” módra.
2. Kapcsoljuk le az imént konfigurált állomást.
3. Végezzük el az első műveletet a másik állomáson is.
4. Indítsuk újra a második állomást.

Ezzel elértük, hogy a fürtszolgáltatás nem indult el, így a lemezeket sem fogja. A szignatúrát a Windows 2000 Resource Kit dumpcfg.exe segédprogramjával nyerhetjük ki. Futtatásához rendszergazdai jogokkal kell rendelkeznünk. Kapcsoló nélkül indítva, két partíciót tartalmazó lemeznél a következő eredményt kapjuk:

```
C:\dumpcfg

[System Information]
Computer Name: TESTSERVER
Cluster name (DNS): TESTSERVER.falatrax.hu
Cluster name (NetBIOS): TESTVSERV
System Root (install directory): C:\WINNT
OS: Windows 2000 Server
Service Pack:

[DISKS]
Disk Number: 0
```

```
Signature: 3137382D

[Volumes]
Volume #1:
Drive letter: C:
Volume Label:
File System: NTFS
Volume Type: Simple Volume \ Logical Drive
Number of members: 1
Member #1: Partition - Disk: 0,   StartingOffset:
32255 bytes,   Length: 2047 MB

Volume #2:
Drive letter: D:
Volume Label:
File System: NTFS
Boot\Boot.ini\System Volume: B00T  SYSTEM
Volume Type: Simple Volume \ Logical Drive
Number of members: 1
Member #1: Partition - Disk: 0,   StartingOffset:
2146830336 bytes,   Length: 2047 MB
```

Miután a szignatúrát lejegyeztük, törölhetjük a RAID vezérlőben a logikai lemezt, és létrehozhatjuk az újat megnövelt területtel. A Windows 2000 szintjén ez azt jelenti, mintha egy új fizikai lemez került volna a rendszerbe. A lemezkezelő beépülőmodult elindítva az operációs rendszer fel is fedezi majd az „új" lemezt, új szignatúrát helyez el rajta és felajánlja, hogy konvertáljuk dinamikus lemezzé. Ez utóbbi műveletet ne engedélyezzük. Elegendő a régi betűjellel ellátni az új erőforrásunkat.

Most már van egy új, nagyobb területtel rendelkező lemezünk, csakhogy eltérő szignatúrával, így a fürtünk bizonyosan nem fog indulni. A dumpcfg azonban újra a segítségünkre lesz. Újra futtassuk le kapcsolók nélkül és hasonlítsuk össze a kapott eredményt az eredeti futtatás kimenetével. Könnyedén meg lehet állapítani, hogy mi volt az eredeti és mi a jelenlegi szignatúra ugyanannál a meghajtónál. Egy újabb parancs kiadásával máris el lehet tüntetni a különbséget:

```
C:\dumpcfg /s <eredetiszignatúra>
<új_lemez_száma>
```

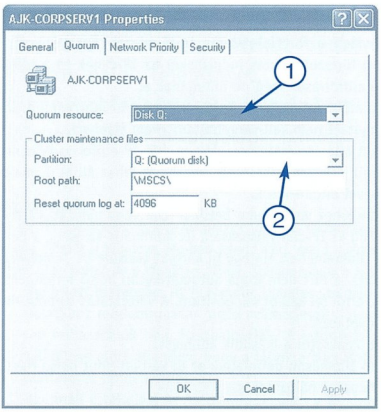
A lényeggel megvolnánk. Most már az eredetivel azonos betűjellel és szignatúrával rendelkezik az új lemezünk, tehát definíció szerint „azonos" is azzal. Nincs más dolgunk, csak a mentés visszaállítása, majd a fürt elindítása a leállításkor alkalmazott módszer visszafelé alkalmazásával.

A fenti módszer egyébként akkor is működik, ha nem tervszerű a lemez cseréje. Meghibásodás után ugyanis a cikk elején említett eseménynapló-bejegyzés pontosan megadja, hogy az eredeti lemez milyen betűjellel és szignatúrával rendelkezett, tulajdonképpen elvégzi helyettünk utólag az adatgyűjtést. A mentést persze semmi sem pótolja!

Mivel a módszer bármely, a fürt által használt lemezre érvényes, akár a quorum-lemezt is cserélhetjük így. Van azonban ennél egyszerűbb módszer is. Íme:

1. Definiáljuk az új lemezt a RAID vezérlőben.
2. Ismertessük meg a diszket a Windows 2000 lemezkezelőjével.
3. Formázzuk meg és adjunk neki betűjelet.
4. Definiáljuk lemezerőforrásként a cluster erőforráscsoportban.
5. Jelenítsük meg a fürtkiszolgáló tulajdonságait. ***(Jobbklikk a fürt nevén->Tulajdonságok)***
6. Adjuk meg az új erőforrást az ábrán 1-gyel jelölt legördülő listában
7. Adjuk meg a megfelelő partíciót a 2-vel jelölt listában, ha több része van a lemeznek.

A fürt elvégzi a megfelelő másolásokat. Ha végzett, próbáljuk ki az erőforrások át- és visszaköltöztetését. Ha nem tapasztalunk hibát, törölhetjük az eredeti quorum-lemezerőforrást.



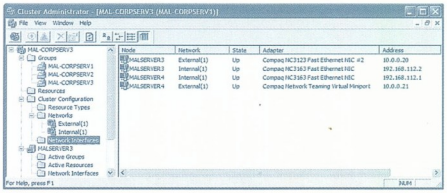
A quorum áthelyezése nem is bonyolult

A fürt IP címeinek változása

A lemezcserénél nem kisebb változás egy fürt életében, ha az állomások hálózati kártyája, vagy azok IP címe változik meg. Körültekintő eljárással azonban ez a feladat is megoldható. A módosulás természetét két részre bonthatjuk: a külső, nyilvános hálózati csatolók vagy IP-címek, és a belső, szívhang forgalomra fenntartott kártyák vagy IP-címek változására.

Kezdjük a nyilvános IP-címekkel. Fontos megjegyzés, hogy a változás közben gondoskodni kell arról, hogy a fürt képes legyen kommunikálni legalább egy tartományvezérlővel. Erre azért van szükség, hogy a fürtszolgáltatás fiókját a tartományvezérlő képes legyen hitelesíteni.

Mozgassuk át valamennyi csoportot a második állomásra, majd az első állomáson változtassuk meg az IP-címet. Indítsuk újra a node-ot. Újraindulás után erről az állomásról indítsuk el a fürtadminisztrátort. Csatlakozzunk újra a „." kapcsolóval az állomáson keresztül a fürthöz. Ellenőrizzük az IP-erőforrásoknál, hogy az új hálózati cím látható-e. Ha igen, mozgassunk át minden erőforrást erre az állomásra, és hajtsuk végre az IP-cím átállítást a másik állomáson is. Újraindítás után győződjünk meg arról, hogy a régi hálózati címek eltűntek.



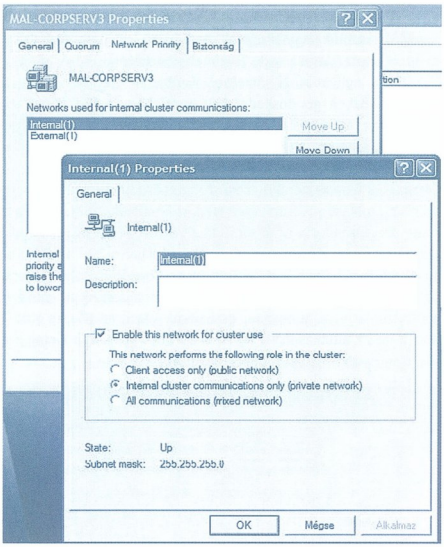
IP címek a fürtben – Itt ellenőrizhető, hogy eltűntek-e a régi címek a váltás után

Ha az IP-cím mellett a hálózati alcímet is meg kell változtatni *(vagy éppenséggel csak azt kell)*, sajnos nem úszhatjuk meg a virtuális szerverek teljes leállása nélkül. Ennek az az oka, hogy a fürtszolgáltatás jelenleg nem támogatja, hogy a fürtöt alkotó állomások külön hálózatban legyenek. *(Ezt egyébként azonos alhálózati maszkkal sem támogatja!)* A megoldás menete tehát a következő:

1. Valamennyi erőforráscsoportban le kell állítani az IP-erőforrásokat és a tőlük függő többi erőforrást is. Ezután lekapcsolható maga a fürtszolgáltatás is. Erre azért van szükség, hogy a fürtszolgáltatás indulásakor az IP-címek és a tőlük függő erőforrások még ne induljanak el.
2. Át kell állítani az állomások IP-címeit a fent leírt módon. Az átmozgatást nem kell végrehajtani ***(nem is lehet)***.
3. Mindkét állomás átállítása után az álló IP-erőforrások paramétereit *(IP-cím, alhálózati maszk)* át lehet állítani, majd egyesével elindíthatók.

Szintén kieséshez vezet, ha a belső IP-cím vagy hálózati kártya változik. Míg az IP-cím változására itt minimális az esély, a belső kártya tönkremehet. Ha egy új kártyát teszünk be *(amelynek új neve van)*, a fürt nem fogja automatikusan belső forgalomra használni, erről fel kell őt világosítani. A változtatás menete röviden a következő:

1. Le kell kapcsolni a második állomást.
2. Az első állomáson definiáljuk az új kártyát mint belső használatra szánt hálózatot.
3. A változások mentése után kapcsoljuk le ezt az állomást is.
4. Indítsuk el a második állomást.
5. Győződjünk meg arról, hogy a fürtszolgáltatás elindult, és a változásokat a futó állomás átvette.
6. Az első node-ot újra indítsuk el. Győződjünk meg arról, hogy a fürtszolgáltatás hiba nélkül működik.



Hálózati csatolók szerepe – tanítani kell a fürtöt...

Tartományváltás

Még ritkábban, de előfordulhat, hogy a fürtöt egy másik tartományba kell áthelyezni. Sajnos itt nagyon kevés jó hírem van. Habár egy fürt áthelyezése nem lehetetlen, a felette futó fürtözött alkalmazások korlátait figyelembe kell venni. Ha Exchange 2000 vagy SQL Servert fürtöztünk, az egyetlen járható út a fürt újraépítése. Más alkalmazások előtt tanulmányozzuk alaposan a gyártó által biztosított dokumentációt. Tegyük fel azonban, hogy csak a Windows 2000 szolgáltatásaiból építettük fel a magas rendelkezésre állású szolgáltatásainkat, és tartományt kell váltanunk. Ekkor a következő tevékenységeket kell elvégezni:

1. Az új fürtszolgáltatás-fiók létrehozása az új tartományban. A cikksorozat második részében leírtak szerint kell eljárni.
2. Le kell kapcsolni a fürtszolgáltatást mindkét állomáson. Az indítási paramétert „kézire" kell állítani.
3. A fürtöt meg kell szabadítani a tartományvezérlő funkciótól. ***(Ez csak Windows 2000 alatt lehetséges. Ha NT4-es fürtünk van, és azok tartományvezérlők, a tartományváltás nem lehetséges.)***
4. El kell végezni az első állomás átmozgatását.
5. Le kell cserélni a cluster zolgáltatás fiókját az új fiókra.
6. El kell indítani a fürtszolgáltatást, és ellenőrizni kell, hogy minden működőképes. Ha minden rendben, újra le kell állítani a szolgáltatást.
7. Át kell mozgatni a másik állomást is.
8. Végre kell hajtani a fiókcserét itt is. Visszaállítható a szolgáltatás indítása automatikusra.
9. Indítható a fürtszolgáltatás.

Ha a DNS szolgáltatásunkat úgy konfiguráltuk, hogy csak biztonságos frissítést engedélyezünk, még egy problémába ütközhetünk. A virtuális szerverek névregisztrációját a korábbi fiók végezte, tehát az új fiók regisztrációs kérelmét a DNS kiszolgáló visszautasítaná. A legegyszerűbb megoldás, hogy a DNS zónaállományból töröljük a virtuális szerverek neveit, majd a hálózati néverőforrásokat újra kell indítani. Ekkor már sikeres lesz a regisztráció.

Lepenye Tamás, MCSE 2000
lepenyet@mal.hu

Q251284	Cluster Server Cannot Start If the Quorum Disk Space Is Full
Q280345	Quorum Drive Configuration Information
Q280425	Recovering from an Event ID 1034 on a Server Cluster
Q243195	Event ID 1034 for MSCS Shared Disk After Disk Replacement
Q267548	Cluster IP Resource Fails After Network Change
Q230356	Changing the IP Address of Network Adapters in Cluster Server
Q279119	Unable to Fail Over the Cluster if Nodes Have Different Subnets
Q269196	How to Move a Cluster Server from One Do-main to Another

Windows Update Corporate Edition

Ez a cikk a Windows Update Corporate Editionről *(WUCE)* szól. Ez az eszköz megkönnyíti az Internetről letölthető Windows javítások vállalaton belüli telepítését. A WUCE a jól ismert Windows Update szolgáltatás továbbfejlesztése, segítségével megoldható, hogy a vállalati számítógépek ne közvetlenül az Internetről, hanem belső, vállalati kiszolgálókról töltsék le a javításokat.

Hogyan történik manapság a javítások telepítése? Rendszeresen ellenőriznünk kell, hogy a Windows Update vagy a Microsoft Security Web helyszínen megjelentek-e újabb javítások. Ha igen, kézzel letöltjük, teszteljük, majd szintén manuálisan, vagy a hagyományos szoftvertelepítő eszközök használatával telepítjük. A WUCE ezt a folyamatot automatizálja. Használatával a Windows számítógépek értesítést kapnak az új frissítésekről *(akár rendelkeznek Internet hozzáféréssel, akár nem)*, és a telepítés is megtörténik.

A WUCE komponensei:

- **Tartalomszinkronizáló szolgáltatás.** A szinkronizáló szolgáltatás egy kiszolgálóoldali összetevő, amely a nyilvános Windows Update szolgáltatástól szerzi be a legújabb frissítéseket. Ha új frissítések kerülnek a webre, azok meghatározott ütemezés szerint automatikusan letöltődnek. Ha nincs ütemezés beállítva, a letöltés kézzel is elvégezhető.
- **Intranetes Windows Update kiszolgáló.** Ez a kiszolgáló virtuális Windows Update kiszolgálóként működik az ügyfélszámítógépek számára, magyarul az Internetes Windows Update helyett ehhez csatlakoztathatók az ügyfélszámítógépek. HTTP protokollon keresztül szolgálja ki a munkaállomásoktól érkező, frissítésekre vonatkozó kérelmeket.
- **A frissítések felügyelete.** A rendszergazda felelőssége, hogy meghatározza, hogy *(szükség szerinti belső tesztelés után)* mely frissítéseket hagyja jóvá telepítésre, és hogy mikor kerüljenek telepítésre. Beállítható, hogy a hálózat mely számítógépei tölthetik le és telepíthetik egy adott vállalti Windows Update kiszolgálóról származó frissítéseket. Ez a felügyeleti lehetőség Active Directory környezetben csoportházirend segítségével, egyéb esetben a rendszerleíró adatbázis használatával érhető el. Teljes egészében a rendszergazdán múlik, hogy a vállalat gépei melyik Automatic Updates kiszolgálóhoz csatlakoznak, onnan melyik frissítéseket töltik le, stb.
- **Intelligens ügyféloldali ügynök.** Az ügyféloldali ügynök az intranetes Windows Update kiszolgáló lekérdezésével állapítja meg, hogy melyik frissítésekre van szükség, ezeket a hátérben letölti, és automatikusan telepíti is.

A Windows Update Corporate Edition és más szoftvertelepítési technológiák

A WUCE alapvető célja a vállalati tűzfalon belül történő Windows Update telepítés. Nem használható egyéb szoftvertelepítési megoldásának felváltására, így nem alkalmazható a Systems Management Server *(SMS)* vagy a csoportházirenden alapuló szoftvertelepítés helyett sem. A Microsoft inkább azon dolgozik, hogy a WUCE az SMS-be is beilleszkedjen, hisz így az összes szoftvert az SMS kihasználásával lehetne telepíteni.

Ügyféloldal

Az ügyfél a Windows Update Automatic Updates technológián alapul, amely először a Windows Me-ben jelent meg, majd jelentős fejlesztések után került be a Windows XP-be. Tulajdonságai a következők:

- **Beépített biztonság:** Az Automatic Updates szolgáltatás használatához rendszergazdai jogok szükségesek, így jogosulatlan felhasználók nem használhatják. A letöltött frissítések telepítése előtt ellenőrzi, hogy a fájlok a Microsoft digitális aláírásával rendelkeznek-e.
- **Azonnali értékelés:** Az Automatic Updates a Windows Update szolgáltatás technológiáinak használatával átvizsgálja a rendszert, és meghatározza, hogy egy adott számítógépre mely frissítéseket lehet telepíteni.
- **Letöltés a háttérben:** Az Automatic Updates a Background Intelligent Transfer Service *(BITS)* használatával tölti le a frissítéseket. Ez az új sávszélesség-szabályozó technológia a Windows XP-ben és az annál újabb operációs rendszerekben található meg, és csak az üresjárati sávszélességet használja, így a letöltés nem zavarja vagy lassítja a többi hálózati forgalmat, például az Internetböngészést. *(Maga a BITS szolgáltatás több szót is megérdemelne, de nincs agyonpublikálva a téma. Egyelőre annyit tudunk, hogy az ABC rendben éppen az Automatic Update mögött áll. Ha valaki többet tud róla, ne fojtsa magába, írjon nekünk!)*

Név	Leírás	Állapot	Indítási típus
Automatic Updates	Enables the dow...	Elindítva	Automatikus
Background Intelligent Transfer Service	Uses idle network...	Elindítva	Kézi

Újabb ismeretlen Windows szolgáltatás: a BITS!

- **Láncszerű telepítés:** Ha egyszerre több frissítés telepítésére kerül sor, és az egyik újraindítást igényel, az Automatic Updates mindet egyszerre telepíti, és csak egyszer indítja újra a számítógépet.
- **Több nyelv támogatása:** Az ügyfél a Windows helyi verziôin is használható.

Kiszolgálóoldal

A WUCE kiszolgáló alapja ugyanaz a technológia, amely a Windows Update Web helyszínen 1998 közepe óta működik. Service Pack 2 vagy annál újabb szervizcsomaggal frissített Windows 2000 Serverre telepíthető. A kiszolgálón engedélyezni kell az Internet Information Services-t *(IIS)*.

Fontos! Mielőtt a kiszolgálót vállalati Windows Update kiszolgálóként telepítjük, ellenőrizzük, hogy futnak-e rajta a legfrissebb Windows biztonsági javítások! A kiszolgáló többek között a következő tulajdonságokkal rendelkezik:

A frissítéseket tároló számítógépen csak rendszergazdai jogokkal rendelkező felhasználók férhetnek hozzá az adminisztrációs weboldalakhoz *(a rendszergazdai felület webalapú)*. A szinkronizálás során a kiszolgálóra letöltődő minden frissítés digitális aláírása ellenőrzésre kerül. Ha az aláírást nem a Microsoft adta ki, a frissítés törlődik.
A vállalati Windows Update kiszolgálóra szinkronizált frissítések nem válnak azonnal, és automatikusan elérhetővé. Mielőtt a frissítés letölthetővé válik, a rendszergazdának jóvá kell hagynia azt.
A frissítések adatbázisa a nyilvános Windows Update kiszolgálóval manuálisan vagy automatikusan szinkronizálható. A frissítési fájlokat jó előre letölthetjük, vagy a Windows Update letöltési kiszolgálók világméretű hálózatából kiválaszthatjuk a földrajzilag legközelebb eső nyilvános kiszolgálót.
Hogy a frissítés Internetkapcsolattal nem rendelkező hálózatban található számítógépeken is elvégezhető legyen, a kiszolgáló lehetővé teszi a tárolt tartalom exportálását, és egy másik vállalati Windows Update kiszolgálóra történő importálását. A frissítéseken lévő digitális aláírás ettől nem sérül meg, mivel gyakorlatilag egyszerű fájlmásolásról van szó.
Bár a kiszolgáló felhasználói felülete angol nyelvű, támogatja a más nyelvi verziójú operációs rendszerekhez készített frissítések közzétételét is. Beállíthatjuk, hogy mely nyelvekre vonatkozó frissítéseket töltse le.
A jóváhagyási és a szinkronizálási műveletek naplózásra kerülnek. A kiszolgáló állapota folyamatosan ellenőrizhető.

A frissítés menete

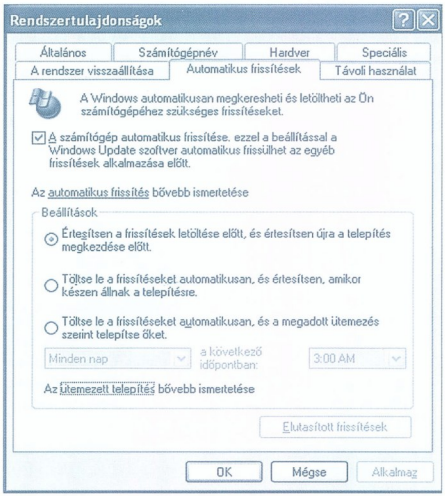
Állítsuk be valamilyen ütemezés szerint az Automatic Updates házirendjét *(például minden nap hajnali háromra)*. A frissítés sikeres letöltése után az Automatic Updates naplózza az eseményt. Ezután a telepítés előtt öt percig visszaszámlálás jelenik meg a képernyőn. Ha a rendszerben van bejelentkezett rendszergazda, ekkor még leállíthatja a telepítést *(így az másnap hajnali háromra kerül át)*. Ha nincs bejelentkezve rendszergazda, vagy ha nem állítjuk le a visszaszámlálást, a telepítés automatikusan folytatódik.

Kiszolgálók frissítése

Míg a munkaállomások akár minden éjjel újraindíthatók, ez a kiszolgálókról már nem mindig mondható el.
Kritikus fontosságú, újraindításra kényes kiszolgálókon úgy állítsuk be az Automatic Updatest, hogy automatikusan töltse le a frissítéseket, és értesítsen, ha a frissítések telepítésre készen állnak. A frissítés sikeres letöltése után az Automatic Updates naplózza az eseményt, hogy a frissítés telepítésre készen áll, de a telepítés nem indul el automatikusan.
Ha távolról ellenőrizzük a kiszolgáló rendszereseményeit, látni fogjuk, hogy a frissítés telepítésre kész. Mi határozzuk meg, mikor aktuális a következő tervezett rendszerkarbantartás. Az ennek megfelelő napon és időben rendszergazdaként bejelentkezünk a kiszolgálóra, és az Automatic Updates használatával telepítjük a frissítést.
Esetleg úgy is beállíthatjuk az Automatic Updatest, hogy ritkábban *(például minden szombaton hajnali három órakor)* hajtsa végre az ütemezett telepítést. Ha ez biztonsági kérdéseket vet fel *(miért pont a kiszolgálókon ér rá egy teljes hétig a legújabb biztonsági javítások telepítése?)*, nincs mese, kézi telepítésre kell felkészülnünk. Az emberi gondolkodás nem mindig spórolható meg.
Az ütemezett telepítések ritkítása nem jelenti a letöltések kimaradását is, a BITS szorgosan töltögeti a telepítenivalót. Ha nem akarjuk azonnal végrehajtatni a telepítést *(bezárjuk az Automatic Updates ablakot)*, az ikon megmarad az értesítési területen, így az ütemezett telepítés előtt is bármikor elkezdhetjük a telepítést. Ha erre szombat hajnali 3-ig nem került sor, automatikusan elindul a munka.
A Windows Update szolgáltatáson közzétett minden frissítést érdemes tesztelni, mielőtt azok a hálózat számítógépeire települnek. Miután teszteltük a frissítéseket, jóváhagyjuk azokat.

A Windows Update a felhasználó nézőpontjából

A felhasználók az Automatic Updates Clientet egy Varázsló használatával állíthatják be, amely 24 órával az után jelenik meg, hogy a számítógép kapcsolatot létesített a Windows Update szolgáltatással. Az Automatic Updates Client beállítása helyileg a Vezérlőpulton található Automatic Updates beállítási oldal használatával *(ez a Windows XP-ben a* **System Properties***-ben; a Windows 2000-ben az* **Automatic Updates Properties***-ben található)*, távolról pedig csoportházirend vagy a rendszerleíró adatbázis használatával is elvégezhető. A **System Properties** beállítási lehetőségeit az alábbi ábra mutatja be:

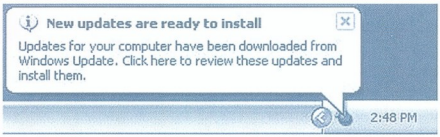


Az Automatic Updates beállítási lehetőségei egy magyar Windows XP-n

A következő lehetőségek állnak rendelkezésre:

- Értesítés a frissítések letöltése előtt, majd újabb értesítés a telepítés előtt.
- A frissítések automatikus letöltése, majd értesítés a telepítés előtt.
- A frissítések automatikus letöltése, majd telepítése a meghatározott ütemezés szerint.

Az értesítést az értesítési területen megjelenő ikon és üzenet végzi. Az események a rendszernaplóban is tárolódnak.



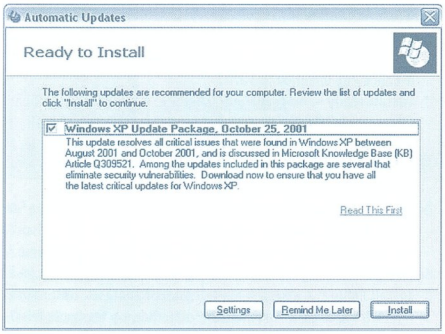
A Windows Update ikonja az értesítési területen

A letöltés

Az Automatic Updates a frissítéseket a felhasználó által meghatározott beállítások szerint tölti le. A Background Intelligent Transfer Service *(BITS)* szolgáltatás használatával a letöltést üresjárati sávszélesség használatával végzi. Ha az Automatic Updates úgy van beállítva, hogy értesítse a felhasználót a letöltésre készen álló frissítésekről, az értesítést a számítógép egy bejelentkezett rendszergazdájának küldi el. Ha rendszergazda nincs bejelentkezve, az Automatic Updates megvárja, amíg valamelyik bejelentkezik, és akkor küldi az értesítést.

A telepítés

Ha az Automatic Updates úgy van beállítva, hogy értesítse a felhasználót, ha egy frissítés telepítésre készen áll, az értesítés a rendszer eseménynaplójába és az értesítési területre kerül.
Ha az üzenetre, vagy az értesítési területen lévő ikonra kattintunk, az Automatic Updates az alábbi ábra szerint megjeleníti a telepítésre készen álló frissítéseket. Ezután az Install gombra kattintva el kell indítania a telepítést. Ha a frissítés telepítéséhez a számítógép újraindítására is szükség van, erről egy üzenet jelenik meg. A rendszer újraindításának elvégzéséig az Automatic Updates nem képes újabb frissítést kezelni.

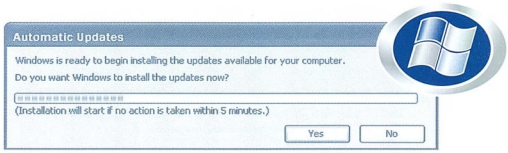


Az Automatic Updates „Ready to Install" párbeszédpanel

A Remind Me Later gomb használatával a telepítés elhalasztható. Választható időtartamok: 30 perc, 1 óra, 2 óra, 4 óra, 8 óra, másnap, 3 nap múlva.

Ütemezett telepítés

Ha az Automatic Updates úgy van beállítva, hogy a telepítést ütemezés alapján végezze, az új frissítések letöltődnek és a telepítésig várakoznak. A rendszergazda az értesítési területen megjelenő ikon segítségével értesítést kap. Ha a felhasználó ekkor az ikonra vagy az üzenetre kattint, egy, a fenti ábrához hasonló párbeszédpanel jelenik meg, amelyen a **Remind Me Later** gomb le van tiltva. Ekkor elvégezhető a telepítés.
Az ütemezésnek megfelelő napon és időben az Automatic Updates telepíti a frissítést és *(ha szükséges)* újraindítja a számítógépet, még ha nincs is helyi rendszergazda bejelentkezve. Ha pedig van, figyelmeztetést jelenít meg, hogy a telepítés el fog kezdődni.



Az Automatic Updates telepítés előtti visszaszámlálása

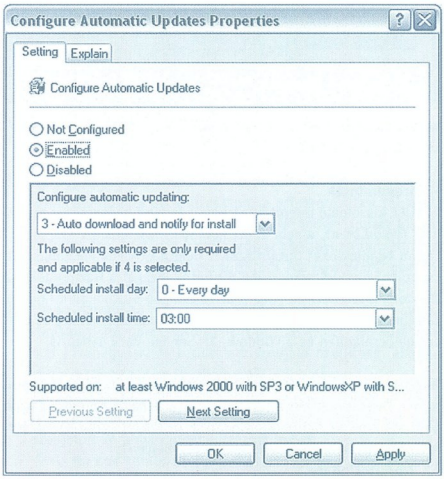
Ha újraindításra is szükség van, és rendszergazda van bejelentkezve, visszaszámlálás jelenik meg, amely figyelmezteti a hamarosan bekövetkező újraindulásra.

Felügyelet házirendek használatával

Mint azt többször említettük,az Automatic Updates működése Active Directory környezetben csoportházirend szabályok beállításával szabályozható. A csoportházirendben meghatározott beállítások mindig elsőbbséget élveznek a felhasználó által beállítottakhoz képest *(ilyenkor az Automatic Updates Vezérlőpultról elérhető beállítási lehetőségei le vannak tiltva)*.

Az Automatic Updates beállítása

Ez a csoportházirend beállítás *(amely a User Configuration\Administrative Templates\Windows Components\Windows Update helyen található)* határozza meg, hogy az adott számítógép kap-e biztonsági frissítéseket és egyéb fontos letöltéseket az Automatic Updatesen keresztül. Ha ezt engedélyezzük, ez egyúttal meghatározza a letöltési és telepítési működést is.



Csoportházirend beállítás az Automatic Updates szolgáltatáshoz

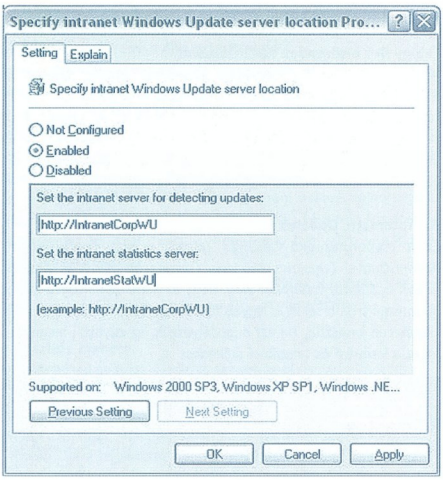
A legördülő menüben a következő három lehetőség közül választhatunk:

- Értesítés mind a letöltés, mind a telepítés előtt.
- A letöltés automatikus, de értesítést kapunk telepítés előtt.
- Automatikus letöltés, ütemezett telepítés. Ha az Automatic Updates ütemezett telepítésre van beállítva, a telepítés napját és idejét is meg kell adni.

Ha ez a házirend le van tiltva *(Disabled)*, az Automatic Updates semmilyen rendszerfrissítést nem végez, a közzétett frissítéseket kézzel kell letölteni és telepíteni a http://windowsupdate.microsoft.com címen található Windows Update helyszín használatával.

A vállalati Windows Update kiszolgáló beállítása

A csoportházirend segítségével egy különálló statisztikai kiszolgáló is beállítható, amely a letöltési és telepítési állapotot figyeli. A statisztikai kiszolgáló egy olyan IIS 5.0, amelyen engedélyezve van a naplózás.



Csoportházirend beállítás a vállalati Windows Update kiszolgáló megadásához

Házirendminták

A WUCE kiszolgáló telepítőcsomagja házirend mintafájlt is tartalmaz *(Wuau.adm)*, amely az itt korábban részletezett csoportházirend beállításokat tartalmazza. E beállítások egy mozdulattal a csoportházirend-szerkesztőbe tölthetők. Ezeket a Házirendeket a Windows 2000 Service Pack 3 javítócsomag System.adm fájlja is tartalmazza, és később megtalálhatók lesznek a Windows .NET kiszolgálócsaládban és a Windows XP Service Pack 1-ben is.

Beállítási lehetőségek Active Directory nélküli környezetben

Active Directory híján a rendszerleíró adatbázis szerkesztésével állíthatjuk be az Automatic Updatest *(a HKLM\Software\Policies\Microsoft\Windows\WindowsUpdate\AU helyen)*:

- **NoAutoUpdate**
  Értéktartomány = 0|1. 0 = az Automatic Updates engedélyezése ***(alapértelmezés)***, 1 = az Automatic Updates letiltása.
- **AUOptions**
  Értéktartomány = 2|3|4. 2 = értesítés a letöltésről és a telepítésről, 3 = automatikus letöltés, értesítés a telepítésről, 4 = automatikus letöltés, ütemezett telepítés. Az értesítést minden esetben a helyi rendszergazda kapja.
- **ScheduledInstallTime**
  Értéktartomány = n; ahol az n = a telepítés ideje 24 órás formátumban ***(0-23)***.
- **UseWUServer**
  Állítsuk 1-re, hogy az Automatic Updates a Windows Update kiszolgálót a WUServer-ben meghatározott módon használja.
- **ScheduledInstallDay**
  Értéktartomány = 0|1|2|3|4|5|6|7. 0 = Minden nap; 1-től 7-ig = a hét napjai vasárnaptól (1) szombatig (7).

A munkaállomások egyedi WUCE kiszolgálóra irányításához új registrykulcsokat kell felvenni a következő kulcs alá:

```
HKLM\Software\Policies\Microsoft\Windows\
  WindowsUpdate
```

- **WUServer**
  A Windows Update intranet kiszolgáló beállítása a HTTP neve alapján (például http://intranetcorpwu).
- **WUStatusServer**
  A Windows Update statisztikai intranet kiszolgáló beállítása a HTTP neve alapján (például http://intranetcorpwu).

Rendszeresemények

Az Automatic Updates a következő rendszereseményeket naplózza:

- Unable to connect: az Automatic Updates nem képes a frissítési szolgáltatáshoz *(a Windows Update-hez vagy a vállalati Windows Update kiszolgálóhoz)* kapcsolódni, és így nem képes a beállított ütemezésnek megfelelően letölteni és telepíteni a frissítéseket. A Windows folyamatosan próbálkozik a kapcsolat létrehozásával.
- Install ready – no recurring schedule: Az eseménynaplóban felsorolt letöltött frissítések telepítésre készen állnak. A frissítések telepítéséhez a rendszergazdának be kell jelentkeznie arra a számítógépre, ahol az ikon megjelent az értesítési területen.
- Install ready – recurring schedule: Az eseménynaplóban felsorolt letöltött frissítések telepítésre készen állnak. Itt a frissítések telepítésének ütemezett napja és ideje is fel van tüntetve.
- Install Success: A sikeresen telepített frissítések felsorolása.
- Install Failure: A sikertelenül telepített frissítések felsorolása.
- Restart required – no recurring schedule: A felsorolt frissítések telepítésének befejezéséhez a számítógépet újra kell indítani. Az újraindítás végrehajtása előtt a Windows nem képes új frissítéseket letölteni.
- Restart required – recurring schedule: A felsorolt frissítések telepítésének befejezéséhez a számítógép öt percen belül újraindul. Az újraindítás végrehajtása előtt a Windows nem képes új frissítéseket letölteni.

A WUCE kiszolgáló beállításai

A beállítási lehetőségek a következők:

- A kiszolgáló neve.
- Annak meghatározása, hogy a frissítések az Internetes Windows Update szolgáltatáson vagy a vállalati intraneten találhatók-e meg. Ha az intraneten, akkor a kiszolgálón IIS 5.0-nek kell lennie és meg kell adni a fájlok helyét.
- A proxykiszolgáló adatai az Internetkapcsolathoz.
- A korábban jóváhagyott tartalom kezelésére vonatkozó beállítások. Ezek akkor fontosak, ha a korábban jóváhagyott tartalom később módosul a Windows Update szolgáltatáson.

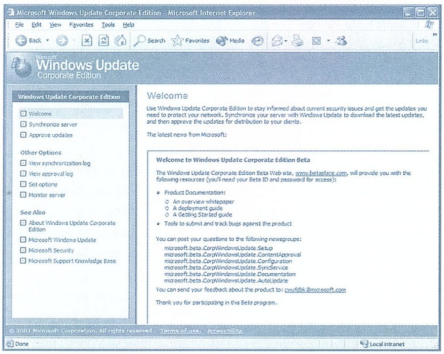
A kiszolgálónév kivételével ezek a beállítások később a HTTP-n keresztül elérhető adminisztrációs felület használatával módosíthatók.

A Windows Update a rendszergazda nézőpontjából

A WUCE kezelésének négy legfontosabb feladata a következő:

1. A kiszolgáló beállítása a telepítés után.
2. A nyilvános Windows Update szolgáltatáson és a vállalati Windows Update kiszolgálón található tartalom manuális vagy automatikus szinkronizálása.
3. A szinkronizált tartalom kiválogatása és jóváhagyása az Automatic Updates ügyfelet futtató számítógépek számára.
4. A kiszolgáló állapotának és naplójának ellenőrzése.

Ezek a kezelési feladatok weboldalak használatával végezhetők el, amelyek magán a vállalati Windows Update kiszolgálón találhatók meg. A Windows Update Corporate Edition ezen verziójához nincs olyan kezelőfelület, amellyel több kiszolgálót egyszerre kezelhetnénk.



A Windows Update Corporate Edition kiszolgáló kezelése

A kezdőlap

A kezelőeszközökhöz tartozó dinamikusan generált kezdőlapon többek között a következők jelennek meg:

- Leírások új vagy frissített biztonsági frissítésekről
- Információk a kiadott javítócsomagokról
- Információk a Windows Update Corporate Edition-höz megjelent frissítésekről

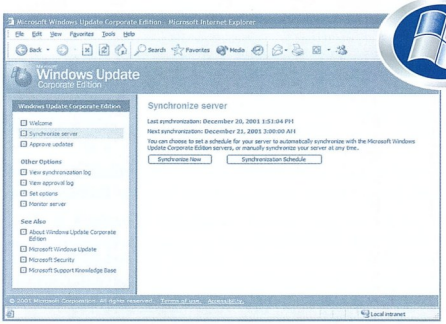
A kiszolgáló szinkronizálása

A Windows Update kiszolgálóról két tartalomtípus szinkronizálható:

- Frissítési metaadatok
- Frissítőfájlok

A frissítési metaadatok tartalmazzák a rendelkezésre álló frissítések leírásait és a frissítések alkalmazási szabályait. A metaadatok minden szinkronizálás során letöltődnek.

A frissítőfájlok azok a fájlok, amelyek egy frissítés jóváhagyása után telepítődnek. Ezek leírása szerepel a metaadatok között. A rendszergazda meghatározhatja, hogy ezek a fájlok a vállalati intranetről vagy a nyilvános Windows Update kiszolgálókról töltődjenek-e le.



A szinkronizálási oldal

A szinkronizálás beállításai

Két lehetőség közül választhatunk:

- Synchronize Now *(manuális szinkronizálás)*
- Synchronization Schedule *(ütemezett szinkronizálás)*

Ha a kiszolgáló úgy van beállítva, hogy a frissítőfájlok az intraneten tárolódnak, a szinkronizálás során a fájlok a frissítési metaadatokkal együtt szinkronizálódnak.

A szinkronizálás folyamata

A szinkronizálás során a vállalati Windows Update kiszolgáló a következő műveleteket végzi:

Interneten keresztül csatlakozik a Microsoft nyilvános Windows Update szolgáltatásához, és tömörített csomag formájában letölti a legfrissebb frissítési metaadatokat.

1. Ellenőrzi, hogy a letöltött csomag tartalmazza-e a Microsoft digitális aláírását. Ha igen, megnyitja a csomagot. Ha nem, a csomag törlésre kerül.
2. Összehasonlítja az új metaadatokat a helyi adatokkal, és kiválogatja az új és a frissített fájlokat.
3. Ha egy korábbi frissítés már nem található meg a nyilvános Windows Update szolgáltatáson, a szinkronizálási folyamat azt a frissítést eltávolítja a vállalati Windows Update kiszolgálóról. Így ezután a kiszolgálóhoz csatlakozó ügyfélgépek már nem tudják a frissítést letölteni ***(az ügyfelekről nem távolítja el a korábban már letöltött frissítéseket)***.
4. Ha egy korábban már letöltött fájl frissítésre került, a szinkronizálási folyamat automatikusan frissíti azt. Ha korábban már jóváhagyásra is került, a kiszolgáló beállításai alapján vagy automatikusan jóváhagyódik, vagy nem.
5. Ha a kiszolgáló úgy van beállítva, hogy a szinkronizálás a frissítőfájlokra is vonatkozzon, az új fájlok letöltődnek a kiszolgálón korábban meghatározott helyre.
   - Ha egy korábban már letöltött fájl nem található meg az intranet helyen, az újból letöltődik.
   - Ha egy, a metaadatokban meghatározott fájl nem tölthető le, az ezzel kapcsolatos metaadatok törlődnek a jóváhagyásra váró frissítéseket tartalmazó listáról.

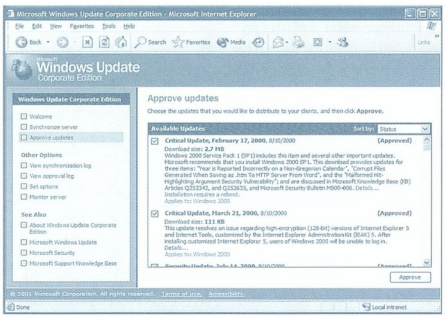
6. Ha a manuális szinkronizálás sikertelen, ez megjelenik a szinkronizálási naplóban is. Ha az ütemezett szinkronizálás sikertelen, az eredetileg ütemezett időpont után 30 percenként a kiszolgáló újra megkísérli a szinkronizálást, amíg a végrehajtás sikeres nem lesz, vagy amíg ki nem kapcsoljuk az ütemezést.

A szinkronizálás után minden esetben szinkronizálási napló készül, amely a rendszergazdai felületen keresztül tekinthető meg.

A frissítések jóváhagyása

Valamelyik rendszergazdának a vállalati Windows Update kiszolgálóra letöltött frissítéseket jóvá kell hagynia ahhoz, hogy a kiszolgáló elérhetővé tegye a hozzá csatlakozó, Automatic Updates ügyfelet futtató számítógépek számára.

A jóváhagyásra az Approve Updates oldal használatával kerül sor.



A frissítések jóváhagyása az Approve oldalon történik

A csomagok adatai

Az Approve Updates oldalon a kiszolgálón megtalálható minden frissítés szerepel. A következő adatok tekinthetők meg: a frissítés neve, kiadásának dátuma, méret, a csomag állapota *(most jóváhagyott, nem jóváhagyott, frissített, új)*, rövid leírás, hivatkozás további adatokhoz, megjegyzés, hogy a számítógépet újra kell-e indítani a csomag telepítése után, megjegyzés, hogy a csomag kapcsolatban áll-e más csomagokkal, és azoknak a Windows operációs rendszereknek a felsorolása, amelyekre a csomag alkalmazható.

A felsorolás dátum, állapot *(New, Approved, Not Approved, Updated, stb.)*, Windows operációs rendszer és név szerint rendezhető.

A Details *(részletek)* hivatkozással a következő adatok érhetők el:

- Az aktuális csomag (.cab/.exe) neve és elérési útvonala, a frissítés telepítési mérete.
- A csomag telepítésekor a parancssorra küldött telepítési paraméterek. A paraméterek használatával a rendszergazda szükség szerint az Automatic Updates-től függetlenül, manuálisan is telepítheti a csomagot.
- A csomag által támogatott nyelvek.
- Egy újabb hivatkozás, amelyre kattintva a Windows Update Web helyszínen további információkat olvashatunk a frissítésről.

A már jóváhagyott frissítések neve mellett a jelölőnégyzet be van jelölve.

Néha egy vagy több frissítés más frissítésekkel is kapcsolatban áll; például a Windows Media Player egy Internet Explorer frissítéssel. Ilyenkor a kapcsolódó frissítéseket együtt kell jóváhagyni. A kapcsolatok a felhasználói felületen is láthatók.

Ha egy más frissítéssel kapcsolatban álló frissítést jóváhagyunk, vagy visszavonjuk annak jóváhagyását, egy figyelmeztető üzenetet kapunk, melyben szerepel a kapcsolódó frissítések neve. Ha elvégezzük a frissítések jóváhagyását, vagy a jóváhagyás törlését, ennek megfelelően a kapcsolódó frissítés is jóváhagyódik, vagy annak jóváhagyása is törlődik.

A jóváhagyott frissítések azonnal elérhetőkké válnak az ügyfélszámítógépek számára, bár ezek a számítógépek esetleg csak később töltik le őket.

A jóvá nem hagyott frissítések nem válnak elérhetőkké a kiszolgálóhoz később csatlakozó számítógépek számára. Ugyanakkor nincs olyan lehetőség, amellyel el lehetne távolítani a frissítéseket azokról a számítógépekről, amelyek korábban már letöltötték az így eltávolított frissítéseket, vagy amelyek éppen a frissítés letöltését végzik.

Minden jóváhagyással kapcsolatos feladat tárolódik a jóváhagyási naplóban, amely a Windows Update Corporate Edition kezelőfelületének bal oldali ablaktábláján keresztül érhető el.

A kiszolgálón elvégzett műveletek és a kiszolgáló állapotának ellenőrzése

Mivel a Windows Update Corporate Edition feladatainak nagy része a frissítések szinkronizálásához és jóváhagyásához kapcsolódik, a rendszergazda két külön naplót *(egy szinkronizálási és egy jóváhagyási naplót)* tekinthet meg. A naplók a kiszolgálón XML fájlban tárolódnak.

A frissítések célszámítógépeken megfigyelhető állapotáról egy információs oldal tájékoztatja a rendszergazdát, mivel ezek az adatok a RAM-ban tárolódnak és így időnként frissíteni kell őket.

Beszerezhetőség

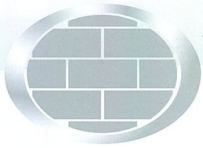
A termék jelenleg nem beszerezhető. 2002. második felében jelenik meg.

A cikkben szereplő URL-ek:

[1] http://www.microsoft.com/security/

[2] http://windowsupdate.microsoft.com/default.htm

Windows 2000 IP Security



Windows 2000 IP Security

Valószínűleg néhány évvel ezelőtt még nem tűnt annyira fontosnak, hogy a hálózaton is biztonságban érezhessük adatainkat. Mi is magyarázhatná különben, hogy rengeteg hálózati protokollból alapvetően hiányoznak azok a biztonsági elemek, amelyek az adatok olvasását, eltérítését, meghamisítását megakadályozhatnák?

A legtöbb általánosságban használt hálózati protokoll *(és ez nem csak a Windows-ra igaz)* a mai napig csak külön ügyeskedés árán *(vagy akkor sem)* képes arra, hogy a felhasználók nevét, jelszavát, vagy az átvitt adatokat védje. Sorolhatnánk a kényes protokollokat *(HTTP, FTP, SMTP, POP3, IMAP4, telnet, NetBIOS SMB, SNMP, DNS, stb.)*, de talán ebből a rövid listából is látszik, hogy a napi munkánk során gyakorlatilag percről percre értékes információkat „teregetünk" ki a hálózatra. Hiába védik például az NTFS jogok a cég kiszolgálóján megosztott fizetések.xls-t, ha egyetlen hálózatanalizátor program segítségével annak a tartalmát a „drótonˮ is láthatjuk – mindössze azt kell kivárnunk, hogy a főnök egyszer megnyissa azt.

Az IPSec protokoll

A Windows 2000 IPSec-implementációja az Internet Engineering Task Force IPSec ajánlásai **[1]** alapján készült. A projekt egyébként a mai napig él; tagjai jelenleg többek között azon dolgoznak, hogyan lehetne az IPSec forgalmat hálózati címfordításon *(NAT-on)* keresztül vinni *(ebből máris kiderül, hogy egyelőre nem lehet; ez a funkció először a Windows .Net Server-ben kerül majd elő)*; illetve azon, hogyan fog működni az egész az új IP protokoll, az IPv6 felett. Az IPSec nyílt ipari szabvánnyá lett, ennek köszönhetően ma már nem csak Windows 2000-ek, hanem IPSec-kompatibilis hálózati eszközök *(például routerek)* között is kialakíthatunk titkosított kommunikációs csatornákat. Cikkünkben a Windows-Windows IPSec csatornákról lesz szó.

Biztonsági algoritmusok

A biztonságos, adott esetben titkosított adatátvitelhez az IPSec különböző kriptográfiai algoritmusokat használ:

- Hash funkciók *(MD5, SHA1)*: a csomagok integritását biztosító algoritmusok. Az IPSec üzemmódtól függően az átvitt csomagok mellé a két algoritmus valamelyikével ellenőrzőösszeget generál; ez biztosítja, hogy a csomag tartalma útközben nem változhat meg.
- Titkosítási algoritmusok *(DES, 3DES)*: a hash funkciók csak a csomagok sértetlenségét biztosítják, tartalmuk ettől még nyilvános. Az átvitt adatokat persze titkosíthatjuk is, ehhez használhatjuk az 56 bites kulcsot használó DES, valamint a jóval erősebb, 3x56 bites kulccsal működő 3DES algoritmust.
- Kulcsmenedzsment: a titkosítási kulcsban azonban a két félnek biztonságos módon meg kell egyeznie; a közös kulcs generálását a Diffie-Hellman (DH) algoritmus segítségével végezhetjük el.

Talán látszik, hogy a konkrét, titkosított IPSec forgalom indulását nem kevés adminisztráció előzi meg *(hogy mást ne mondjunk, például meg kell egyezni a használt algoritmusokban, kulcsméretekben, és persze magában a titkosításhoz használt kulcsban is)*. Ezt a megállapodást nevezik Security Association-nak, röviden SA-nak.

Azonosítási módok

Az IPSec kommunikációban részt vevő gépek *(figyelem, nem a felhasználók!)* még a csatorna felépítése előtt kölcsönösen azonosítják egymást. A Windows háromféle azonosítási módot kínál:

- Kerberos V5: a gépek azonosítása a Kerberos protokoll segítségével. Ennek feltétele, hogy a gépek mindegyike Windows tartományi tag legyen *(mégpedig nyilván közös, egy fába tartozó, vagy egymásban bízó tartományoké)*. A működés értelemszerű feltétele továbbá, hogy a csatorna felépítésekor legyen elérhető, működő tartományvezérlő.
- Public Key Certificate: nyílt kulcsú *(PKI)* tanúsítvánnyal történő azonosítás. Ehhez a számítógépeknek kell saját PKI tanúsítvánnyal rendelkezniük, mégpedig olyanokkal, amelyek kiadója ***(CA)*** a másik oldal megbízott kiszolgálói között ***(erről részletesen majd a beállításoknál lesz szó)***
- Pre-Shared Key: szöveges *(jelszó, bár inkább jelmondat)* alapú azonosítás, ami a kommunikáció során nyíltan nem jelentkezik, de az adminisztratív felületen, illetve a registryben kódolatlanul látható. Éppen ezért ez a módszer inkább átmeneti célokra alkalmas – bár ez a legegyszerűbben munkára bírható üzemmód.

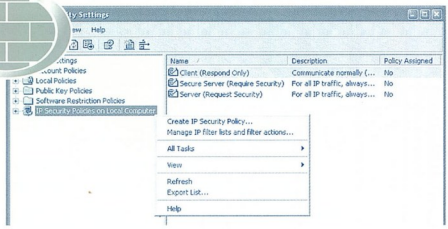
Az IPSec házirend elemei

Az IPSec működését meghatározó szabályrendszert IPSec házirendnek *(IPSec Policy-nek)* nevezzük. Az IPSec házirendet felépítő elemek a következők:

- Szűrőlisták *(Filter Lists)*: szűrőket tartalmazó listák, amelyek az „elfogni" *(titkosítani)* kívánt hálózati kommunikációt definiálják *(például feladó vagy címzett IP címe, a használt portcím és a csomag haladási iránya szerint)*
- Akciók: a szűrőkkel elfogott csomagokkal való tennivalókat határozzák meg. Akció lehet a csomag eldobása *(Block)*, titkosítás nélküli beengedése *(Permit)*, valamint a titkosított csatorna felépítése *(Negotiate Security)*, meghatározott biztonsági paraméterek alapján
- IPSec Policy objektumok: az egyes szűrőlistákat és akciókat összerendelő házirendobjektum; bár több IPSec Policy-t definiálhatunk, azok közül mindig csak egyet rendelünk hozzá ***(Assign)*** a számítógéphez

Az IPSec házirend kezelése

Az IPSec házirendet definiálhatjuk lokálisan *(a Local Security Policy segítségével)* és a Group Policy-n keresztül egyaránt. Mint minden más esetben, tartományi tag Windows 2000-ek és XP-k esetén a Group Policy-ben definiált beállítások *(ha vannak)*, felülbírálják a számítógép helyi beállításait.

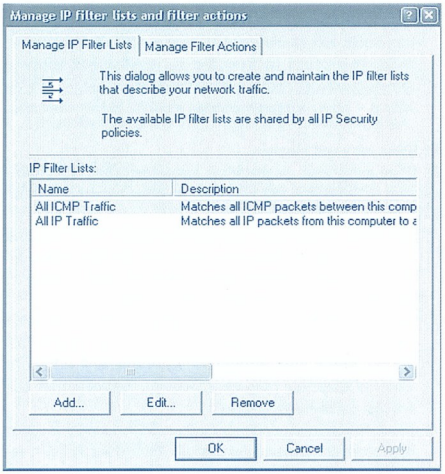


Az IPSec beállításai a helyi biztonsági házirendben

A továbbiakban a helyi biztonsági házirend beállításait használjuk *(tartományon kívüli gépek esetén egyébként mást nem is tehetünk)*.

IPSec szűrőlisták létrehozása

A házirendek létrehozásának első lépése a szűrőlisták definiálása. Ezeket a szűrőlistákat *(és a következő fejezetben tárgyalt akciókat)* nem házirendenként, hanem azoktól függetlenül hozzuk létre, hogy azután a házirendek ezekből, mint építőkockákból táplálkozzanak. Nem véletlen tehát, hogy a szűrőlisták és -akciók definiálására szolgáló menüpontot *(„Manage IP filter lists and filter actions")* nem a „gyárilag" létrehozott policy-k között, hanem bal oldalon, a fában az IP Security Policies sorra jobb gombbal kattintva megjelenő menüben találjuk meg.



A szűrőlisták párbeszédpanelje

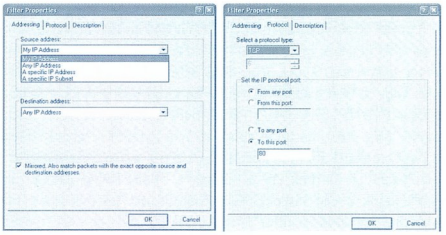
A szűrőlisták között két előre definiált listát is találunk:

- All ICMP Traffic: minden ICMP forgalomra érvényes szűrő, a saját gépünk és bármely más IP cím között. Az ICMP protokollt a ping, tracert és egyéb diagnosztikai parancsok használják.
- All IP Traffic: minden IP hálózati forgalomra érvényes szűrő *(kivéve technikai okokból a broadcast, multicast, Kerberos, RSVP és az IKE protokollt)*; a saját gépünk és bármely más IP cím között.

Ha ezek a szűrőlisták nem eléggé specifikusak *(valószínűleg nem lesznek azok)*, saját szűrőlistákat hozhatunk létre, ehhez a felsorolás alatt kattintsunk az Add... gombra. Ekkor új szűrőlistát készítünk. Adjunk nevet a szűrőlistánknak *(pl. „Webes protokollok")*, majd a következő Add... gomb segítségével hozzuk létre a szűrőlistát alkotó szűrőket *(mondjuk a TCP 80-as és 443-as porthoz)*. Minden szűrő az alábbi feltételeket tartalmazhatja:

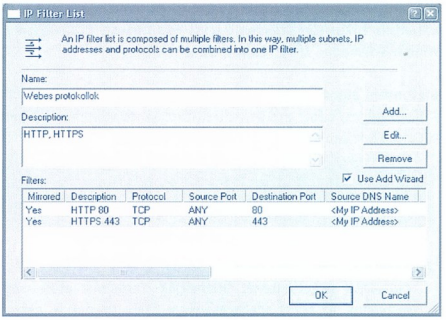
- A csomag feladójának IP címe *(Source IP Address)*: bármely saját, bármely idegen, egy adott IP cím, vagy adott IP címtartomány
- A csomag címzettjének IP címe *(Destination IP Address)*: mint az előző
- A csomag protokollja (Protocol type): ICMP, UDP, TCP, vagy bármely más IP protokoll
- A feladó port címe
- A címzett port címe

Természetesen a protokollok, portcímek megadása nem kötelező, azaz definiálhatunk olyan szűrőt is, amely például két gép *(IP cím)* között mindennemű forgalomra érvényes.



Egy szűrő tulajdonságlapja

Vegyük észre a bal oldali ábra alján a „Mirrored..." feliratot. Az IPSec szűrők minden egyes csomagra külön érvényesek. Ha egy kétirányú kommunikációra érvényes szűrőt szeretnénk létrehozni *(márpedig a legtöbb kommunikáció ilyen, például a böngésző és a webszerver között: kérés-válasz)*, akkor a kimenő és bejövő csomagokra külön-külön szűrőket kellene létrehoznunk (hiszen a feladó IP címe akkor a címzett lesz, a portcím ugyancsak, stb.). Ezt a „felesleges" munkát spórolja meg nekünk az IPSec: ha a „Mirrored" kapcsolót beállítjuk, az általunk létrehozott egy szűrő valójában kettő lesz *(bár ez a grafikus felületen külön nem látható)*.



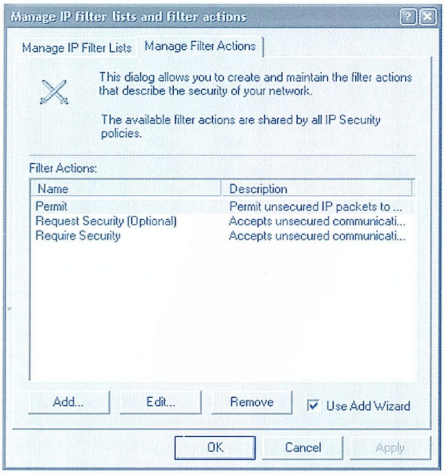
Az általunk létrehozott szűrőlista, benne két tükrözött (azaz összesen négy) szűrővel

Az ábrán láthatjuk, hogy az egyes szűrőknek saját „nevet" adtunk *(a Description oszlopban: HTTP 80, HTTPS 443)*. Ezt az értéket a szűrő tulajdonságlapjának – előbb be nem mutatott – Description oldalán lehet megadni, és érdemes is kitölteni, mert a szűrő neve később megkönnyíti a diagnosztikai munkát *(külön-*

ben csak NoName szűrőkkel dolgozunk és az megnehezítheti a szűrő azonosítását).

Szűrőakciók kezelése

A következő kérdés az, hogy mi történjen a szűrőlistákkal kihalászott csomagokkal. Ezt mindig a házirendben az adott szűrőlistához rendelt szűrőakció határozza meg. Következő dolgunk tehát a megfelelő szűrőakciók létrehozása. Ehhez a „Manage IP filter lists and filter actions" párbeszédpanel második oldalára kattintsunk:



A szűrőakciók párbeszédpanelje

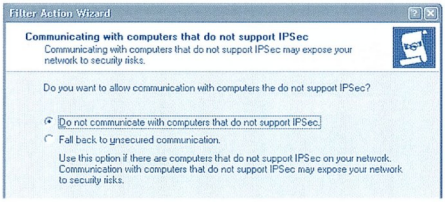
A szűrőlistákhoz hasonlóan itt is találunk néhány „gyárilag" definiált szűrőakciót:

- Permit: titkosítás nélküli forgalom engedélyezése
- Request Security *(Optional)*: titkosított csatorna felépítése az alapértelmezett beállításokkal, vagy ha ez nem sikerül, kommunikáció titkosítatlanul
- Require Security: titkosított csatorna felépítése az alapértelmezett beállításokkal, vagy ha ez nem sikerül, a csomag eldobása

Nincs előre létrehozva, de definiálható olyan akció is *(Block)*, ami a csomagok minden további nélküli eldobását jelenti. Próbaképpen hozzunk létre egy ilyen akciót: kattintsunk az Add gombra, erre elindul a Filter Action Wizard. Adjunk egy hangzatos nevet az akciónknak *(például: Block)*, majd a Filter Action General Options oldalon válasszuk a Block akciót. Ezzel készen is vagyunk. Talán sejti már az Olvasó, hogy a komolyabb kihívást a titkosított csatorna kiépítésére vonatkozó beállítások megadása jelenti. Definiáljunk egy ilyen szűrőakciót, és a varázsló segítségével haladjunk végig a beállításokon. Kattintsunk tehát a szűrőakciók listája alatt újra az Add... gombra, majd az akciónknak adjuk a „Security 1" nevet. A Filter Action General Options oldalon ezúttal válasszuk a „Negotiate security:" *(biztonságos csatorna kezdeményezése)* opciót.

„Fail back to clear"

A varázsló következő oldalán megjelenő kérdés azt firtatja, hogy mi történjen akkor, ha mi ugyan megpróbáltuk a titkosított csatorna kiépítését, de az bármilyen probléma miatt nem sikerült:



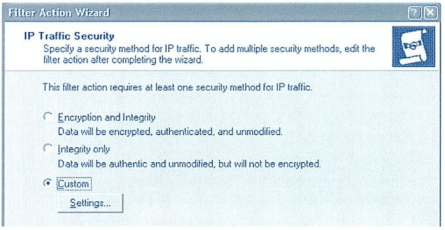
Mi történjen, ha a „túloldal" nem tud IPSec-ül?

- „Do not communicate...": ha nincs IPSec, nincs kommunikáció
- „Fail back to unsecured communication": ha nem sikerül IPSec-kel, a kommunikáció titkosítatlanul folytatódik

Ez utóbbi opciót választva a csomag végül titkosítatlanul jut el a címzetthez, ezért ez nyilvánvaló biztonsági problémákat vethet fel. Cserébe viszont akkor sem veszítjük el a kapcsolatot a külvilággal, ha az IPSec valami miatt nem tért magához. A titkosítatlan kommunikáció során egyébként az IPSec alrendszer adott időközönként *(lásd később)* újra megpróbálkozik a titkosított csatorna felépítésével, és ha sikerült, át is tér annak használatára.

A biztonsági algoritmusok

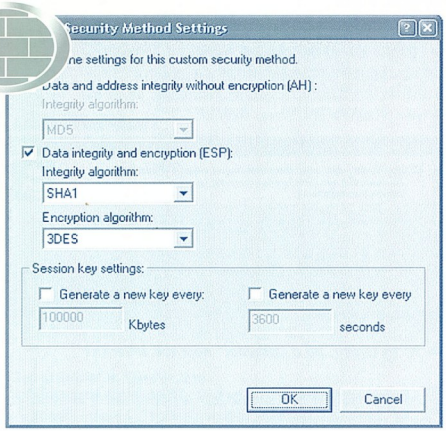
A következő oldalon nyilatkoznunk kell a használni kívánt biztonsági algoritmusokról. Tudatosan nem titkosítási algoritmust írtam, ugyanis az IPSec-nek van olyan üzemmódja is, amikor a csomagok tartalmát nem titkosítjuk, csak azok sértetlenségét garantáljuk.



A szűrőakció biztonsági metódusai

A választható lehetőségek:

- Encryption and Integrity *(Windows 2000-en: High)*: a csomagok titkosítása az alapértelmezett beállítások szerint – az IPSec az ESP *(Encapsulated Secure Payload)* protokollt használja majd
- Integrity only *(Windows 2000-en: Medium)*: csak integritásellenőrzés, titkosítás nélkül – a használt protokoll ekkor az AH *(Authenticated Header)* lesz
- Custom: testreszabott beállítások

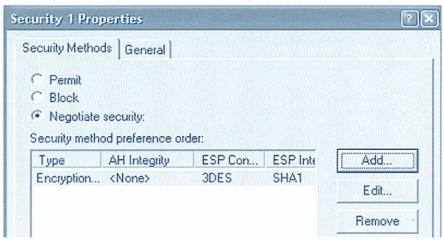


Testreszabott biztonsági paraméterek

A testreszabás során meghatározhatjuk a használni kívánt protokollt *(ESP esetén nincs értelme külön az AH-t is bekapcsolni, mert az ESP – mint láthatjuk – önmaga is tartalmaz integritás-ellenőrzést)*, a titkosítás biztonsági algoritmusait, sőt, még a szakaszkulcsok újragenerálásának küszöbértékeit is módosíthatjuk. *(Az alapértelmezés szerint új titkosítási kulcs készül minden 100 megabájt, illetve 60 perc kommunikáció után)*.

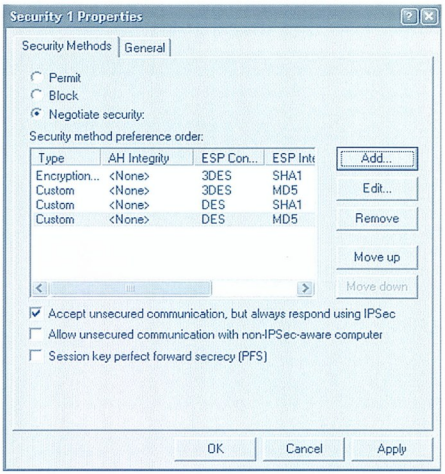
A szűrőakció tulajdonságlapja

A varázsló ezután befejezi a munkáját és létrehozza a szűrőakciót. A további beállításokhoz válasszuk ki a listából a kívánt akciót, majd kattintsunk az Edit... gombra.



A varázsló által beállított biztonsági paraméterek

Láthatjuk, hogy a biztonsági paraméterek listájában az az egy sor szerepel, amit a varázsló futtatása során megadtunk. Itt további beállításcsomagokat hozhatunk létre arra az esetre, ha a „túloldal" valamilyen oknál fogva az elsődleges titkosítási beállításokat nem tudná teljesíteni *(például mert nem képes 3DES titkosításra, satöbbi)*. Az Add... gomb megnyomására megjelenő dialógusablak teljesen hasonló a varázslóban már látottakkal.



Testreszabott biztonsági paraméterek

Ha egynél több biztonsági metódust definiáltunk, azok sorrendjét a Move up/down gombok segítségével módosíthatjuk is. Ha a listában felülre kerülnek a legerősebb titkosítást jelentő beállítások, majd a titkosítási szint lefelé haladva egyre gyengül, biztosak lehetünk benne, hogy a csatorna felépítése során a lehető legbiztonságosabb beállítások fognak érvényesülni.

Folytatjuk...

Fülöp Miklós
mick@netacademia.net

A cikkben szereplő URL-ek:
[1] http://www.ietf.org/html.charters/ipsec-charter.html
[2] http://www.ietf.org/rfc/rfc2409.txt

Digital Encryption Standard

Ritkán megjelenő kriptográfia-sorozatomban a tavaly decemberi nyílt kulcsú titkosítás *(RSA-algoritmus)* után most a szimmetrikus titkosítások legelterjedtebb képviselőjét, a DES *(Digital Encryption Standard)* eljárást nézzük meg értő szemmel. Mire a cikk végére érünk, mindenki tudni fogja, miként működik egy szubsztitúciós-permutációs hálót használó, CBC módú Feistel-cipher.

Messziről szeretném felvezetni a DES-t, mert korántsem olyan tiszta eljárás, mint az RSA. Sőt, első ránézésre kifejezetten zavaros. Ez talán nem is meglepő, ha figyelembe vesszük, hogy pont erre, adatok összezavarására hozták létre. Az IBM saját, Lucifer nevű eljárásának továbbfejlesztéseként a múlt század 70-es éveinek elején, az NSA-val karöltve *(National Security Agency)* kezdett a fejlesztésébe, s később *(1977-ben)* az ANSI szabványtestület X9.32 sorszámmal lajstromba is vette. A fejlesztés alapjául szolgáló Lucifer rendszert Feistel dolgozta ki, s 1974-ben szabadalmaztatta az IBM. Mára a DES összes szabadalma lejárt.

Leírásomban itt-ott kitérek a kriptoanalízis *(kódvisszafejtés)* egy-egy módszerére is, hogy lássuk, mi ellen is kell védekeznünk. A DES használatát húsz évig támogatta az USA kormányzata, s csak az után mondtak le róla, hogy napjaink PC-ivel kevesebb, mint egy nap alatt fel lehet törni. Nem matematikai trükk, hanem az idő törte meg: 64 bites *(a paritásbitek miatt valójában 56 bites)* kulcshosszúságával ma már nyers erővel *(brute force)* is „vígan" törhető, hisz mit nekünk 2^{55} darab próbálkozás!

Kriptoanalízis I. – brute force. A lehetséges titkosítási változatok, az összes kulcs végigpróbálása. A kulcshossz növelésével ez a feltörési módszer egyre több és több időt vesz igénybe, bár láttuk: ami a 70-es években lehetetlen feladatnak tűnt, azt ma már asztali PC-vel lazán megoldjuk.

Ennek ellenére továbbra is ő a szimmetrikus titkosítások legjobbika, leszármazottait *(DESX és 3DES)* használjuk napjaink titkosítási feladataira. *(Ha például megnézzünk az IPSec beállításai lehetőségeit, DES és 3DES szimmetrikus algoritmusok közül választhatunk – más nincs is!)*

Elsőként szögezzük le: a komoly titkosítási eljárások mindegyikének, így a DES-nek is nyilvános az algoritmusa, a titkosító programkód ismerete nem elég a titkosítás feltöréséhez. Kell egy kulcs is. Minden olyan titkosítás, melynek nem hozzák nyilvánosságra a kódját, a saját sírjába esik bele abban a pillanatban, hogy kereskedelmi forgalomba kerül, mert teljesen bizonyos, hogy valaki visszafejti a gépi kódot. így járt például a LanMan „titkosítás" *(hash)*, melybe mindenféle ravasz*(nak hitt)* csűrcsavar, magic number került. Disassembler segítségével minden tekervény kibogozható, s az algoritmus meztelenül áll előttünk. Az algoritmus eltitkolására épített biztonsági megoldások előbb-utóbb egy az egyben szemétdombra kerülnek.

S most lássunk néhány titkosítási módszert, hogy megértsük, miért azt csinálja a DES, amit. Az egyszerűség kedvéért egyelőre szöveges üzenetek kódolásában gondolkodjunk, és tegyünk úgy, mintha az ABC 26 betűből állna *(az angol ABC annyiból áll)*. Kétféle titkosítási elvet nézünk meg, a helyettesítő *(szubsztitúciós)* és a cseberélő *(permutáló)* titkosítást.

1. Helyettesítő titkosítási módszerek

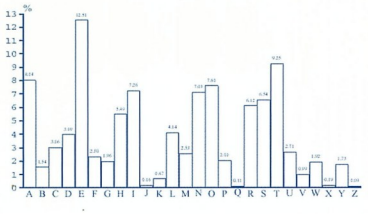
A DES megértéséhez érdekes módon Julius Caesarig kell visszamennünk az időben. A legenda szerint ő találta fel az egyszerű eltolásos titkosírást, ahol a rejtjelezett szöveget az ABC-ben néhány betűvel eltolva írták le. Ez akkoriban elegendőnek bizonyult, a futár útközben amúgy is csupa írástudatlannal találkozhatott. Az eltolásos módszer hátránya, hogy az első néhány betű kitalálása után a többi adja magát, a „rejtjelezés" szinte magától kinyílik. A „brute force" módszer is csak 25 változat végigpróbálását igényli, hisz maximum 25 karakterrel tolható el minden betű. Ez a probléma 1500 éven keresztül senkit sem zavart. A felvilágosodás korában azonban, ahogy egyre több írástudó élőlény jelent meg az evolúció sodrában, felmerült az eredeti, római titkosítás bonyolításának gondolata, hogy mégse minden jöttment legyen képes titkosírást visszafejteni. Megszületett az a helyettesítéses módszer, melynek lényege egy „alternatív" ABC használata. Röviden szólva: egy táblázat alapján minden betűt kicserélünk egy másikra. Az alábbi táblázat a második sorában egy ilyen alternatív ABC-t tartalmaz. Titkosításkor az eredeti szöveg minden betűjét kicseréljük a táblázatban alatta találhatóra. Így lesz a HIBA szóból OZTQ.

A	B	C	D	E	F	G	H	I	J	K	...	Z
Q	T	E	F	S	A	M	O	Z	X	U		I

Alternatív ABC-s, helyettesítő titkosítás

Julius Caesar titkosításának még csak 26 kulcsa volt. ABC-helyettesítő táblázatból viszont már 26! *(faktoriális)* készíthető, így a kulcsok száma 403291461126605635584000000. *(Hotta kedvéért nevezzük nevén: 26! darab permutációja létezik az angol ABC-nek.)* Ennek nem érdemes nyers erővel nekimenni. Ma persze semmit sem érne ez a fajta „titkosítás", mert statisztikai módszerekkel két másodperc alatt kideríthető, melyik betű minek felel meg.

Kriptoanalízis II. – gyakoriságfüggvények. Minden titkosítási módszer, mely megőrzi az eredeti betűk gyakoriságát, hihetetlenül hirtelen feltörhető. Ennek az az oka, hogy az egyes nyelvekben a betűk előfordulási gyakorisága szinte változatlan. Hiába írunk minden E helyett S-et, ha a titkosított szövegben hemzseg az S, tudhatjuk, hogy az az E betűt jelöli. Nem is beszélve a gyakori betűkapcsolatokról: AZ, CS, SZ, ÉS. Ezek segítségével további betűket kapunk, és szép lassan összeáll a teljes szöveg, akár egy keresztrejtvény megfejtése.



Az angol nyelv gyakoriságfüggvénye

Az Enigma

A következő nagy ugrás a második világháború alatt következett be. Ebben a háborúban a rádió vált az elsődleges kommunikációs, csapatirányító technológiává. Nem sokra megyünk füstjelekkel és zászlólengetéssel, ha az irányítani kívánt hajó éppen kétszáz méterrel a tenger szintje alatt található. A rádiójeleknek azonban van egy nemkívánatos tulajdonságuk: irányíthatatlanok. Az adás sajnos „publikus". Kézenfekvő a megoldás: titkosítani kell a mondanivalót! Igen ám, de hogyan?

Olyan titkosításra van szükség, mely nem őrzi meg az eredeti szöveg betűinek eloszlását! A XX. Század elején született, „forgótáras" titkosítógépek utódja az Enigma, mely kereskedelmi forgalomban is kapható volt! A működés lényege röviden a következő: a rejtjelezés során **minden egyes betűre** más-más alternatív ABC-t használunk! Vagy más megközelítésben: Julius Caesar módszerét úgy fejlesztjük tovább, hogy minden egyes betűt más-más mértékben tolunk el az ABC-ben. A titkosítás kulcsa tehát egy eltolási számsor, például: +5, -8, +11, +4, -14, +6 stb. Tehát az üzenet első betűjét öttel előretoljuk, a másodikat az ABC-ben nyolccal előtte lévővel helyettesítjük stb. Láthatjuk, hogy ez a titkosítás eltörli az eredeti betűk gyakoriságát, hisz a számsor kényének engedelmeskedve akár minden E betű különböző cserebetűkre képeződik le. Ha jó a titkosítóalgoritmus, az eltolási számsor véletlenszerű, és ismétlődéseket nem tartalmaz. Mi tehát az Enigma? Egy véletlenszám-generátor! Egészen pontosan pszeudo-véletlen-generátor, egy alapállapotból mindig ugyanazokat az eltolási számokat sorolja fel. Ez a tulajdonság a visszafejtéshez is nélkülözhetetlen. A titkosítás menete a következő: az adó és a vevő megállapodnak saját titkosítómasináik alapállásában *(három ABC-tárcsa és néhány banándugó beállításában)*, majd tekerik a kart, és az Enigma minden bemeneti karaktert más-más eltolással ad vissza.

A németek éveken át sikeresen használták az Enigmát, mivel amint tudomásukra jutott, hogy az ellenség képes feltörni a titkosítást, mindig változtattak rajta valamit *(plusz egy tárcsa stb.)*. Valahányszor a lengyel *(még Lengyelország lerohanása előtt)*, vagy a brit kódtörők eredményre jutottak az Enigma ellen, mindannyiszor rendelkezésükre állt egy falatka eredeti szöveg, s annak titkosított változata. Ebből ki tudták találni, hogy milyen indítóállapota lehetett a gépnek, s így az üzenet egészét is el tudták olvasni. Sőt, mivel a németek naponta csak egyszer dugdosták át a banándugókat, aznap minden üzenetet képesek voltak dekódolni. Volt időszak *(1939, ha jól tudom)*, amikor a lengyelek 85%-os sikerrel olvasták a németek adásait.

Kriptoanalízis III. - ismert szöveg megfejtése. Ez a módszer a brute force *(nyers erő)* alesete. Ha van egy hosszú titkosított szövegünk, melyből bizonyos részeket ismerünk, az ismert adatok birtokában megállapítható a kulcs. Egyszerű: ha tudom, minek kell kijönnie, addig próbálkozom a kulcsokkal, amíg célt nem érek. S ha megvan a kulcs...!

Az Enigma első megbuktatásában jelentős szerepe volt a német nyelvnek, melyben hemzseg az EIN betűkapcsolat. *(főnevek: Einstein, Rosenstein stb, és ein, mint szám)*. A lengyelek készítettek egy 150 ezer elemű EIN-szótárat, és a lehallgatott szöveg minden tripletjét összevetették az előre kikódolt változatokkal. Ennyire egyszerű. Csakhogy akkor még nem volt számítógép! Puszta kézzel csinálták! Az Enigma teljes története az **[1]** címen olvasható.

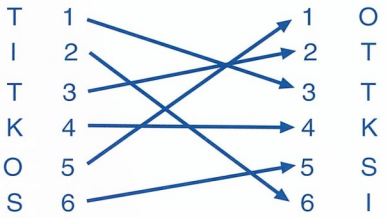
...vissza a jelenbe

De mi köze ennek a DES-hez? Minek ismerni ezeket az ősi, csotrogány titkosítási módszereket? Egyszerű a válasz: mert a DES ezekből a módszerekből építkezik. Igenis használ egyszerű helyettesítéseket. Az úgynevezett S dobozok valójában alternatív ABC-k! Lásd később!

vissza a múltba...

2. Cserberélő *(transzpozíciós, permutáló)* titkosítási módszerek

Julius Caesarnak volt mégegy titkosítási módszere: a betűk sorrendjének összekavarása. Az elsőt felcseréljük a hetedikkel, a negyedikből lesz az első stb. Ehhez segédtáblázatokat kellett készíteni, melyek megmutatták a betűkeverés menetét. Az alábbi ábrán a TITKOS szót titkosítjuk:



Permutációs titkosításhoz készített segédtábla

Hat karakteres szavak 6! = 720 féleképpen titkosíthatók, a fenti háló tehát egy a 720 féle kulcs közül. E titkosítás kulcsa az adott permutációs táblázat, amit így is ábrázolhatunk:

1	2	3	4	5	6
3	6	2	4	1	5

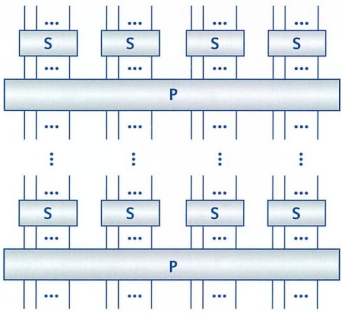
Vagy még rövidebben: 3,6,2,4,1,5. Ha ebben a sorrendben olvassuk a titkosított üzenet betűit, az eredeti szöveget kapjuk vissza. Erre a számsorra kell emlékezni, amit Julius Caesar különböző trükkös versikék, aforizmák bemagolásával tett elviselhetőbbé futárai számára. *(Tehát a futár egy titkosított üzenetet és egy versikét vitt magával. Csak arra kellett ügyelni, hogy a futár ne tudja a versike értelmét, így az ellenség sem tudta kiverni belőle a kulcsot.)*

Block cipher, stream cipher

A helyettesítő módszerek *(Enigma és társai)* az úgynevezett stream, vagy folytonos titkosítások közé tartoznak, mert a titkosítandó szöveget folytonosan, betűről betűre *(vagy akár bitről bitre)* dolgozzák fel, s az algoritmus a végtelenségig futhat. A sorrend-cserebere, vagy permutáló módszerek azonban nem értelmezhetők végtelen hosszúságú szövegeken. A feladat elvégzéséhez tudnunk kell, hogy hány betűnk lesz, vagy ha nem tudjuk, a szöveget fel kell tördelnünk adott méretű **blokkokra**. A DES úgynevezett blokktitkosítás *(a blokkméret 64 bit)*, tehát feltehetőleg van benne cserebere. Van bizony!

Összetett titkosítások

Bármilyen gyatrának tűnnek az eddig felsorolt módszerek, e két forma közül választhatunk. Vagy ABC-n belüli *(helyettesítés)*, vagy szövegen belüli *(transzpozíció)* permutációt alkalmazunk. A DES tervezői a következő feltételezéssel éltek: ha több, egyenként gyenge titkosítást egymás mögé fűzünk, az eredmény egy sokkal erősebb titkosítás lesz. Okosan megtervezett esetben. Ha ugyanis egy +2-es szubsztitúció után csatolunk egy -2-est, nemhogy erősödne a titkosítás, hanem megszűnik. Az egymást gyengítő láncok elleni védekezés jegyében a DES-ben felfűzött, egymás után álló titkosítások különböző típusúak. Nevezzük nevén: a DES egy szubsztitúciós-permutációs hálózat, amely felváltva helyettesíti, majd csereberéli a „betűket" *(a DES bináris titkosítás, tehát bitekkel dolgozik)*. Az alábbi ábrán az S-sel jelölt dobozok az inputon rém egyszerű helyettesítést végeznek, a P-vel jelölt tégalalpok pedig összekeverik az egyes bájtok pozícióit.



xycoi é 48z 8zw9p7r

Szubsztitúciós-permutációs hálózat

Egyszerű nemde? Ha ezt Julius Caesar látná! Az S-helyettesítések és a P-permutáció **kötött táblázatok** alapján történnek *(nyolc darab S-box és egy P-táblázat)*, hogy a DES minél könnyebben hardverre ültethető legyen. S hogy az egyenként elégtelen kódolásokból mégiscsak erős titkosítás szülessen, a DES 16-szor végzi el egymás után az – egyébként egyforma – S és P lépéseket.

Hopp! De hol jön be a képbe a titkosítási kulcs? A permutáció bemutatásánál szóbakerült, hogy a kulcs nem más, mint egy adott permutációs táblázat *(a lehetséges kismillió közül)*, amit a futár versikében megtanult. Itt viszont egyetlen, **kötött permutációs táblázattal** van dolgunk, ha tetszik, ha nem, ezt a táblázatot használja a P lépés:

P			
16	7	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	24	14
32	27	3	9
19	13	30	6
22	11	4	25

A DES permutációs táblája. Erre alkoss versikét!

E közül az egyetlen táblázat közül egyféleképpen lehet választani: ezt választom! Itt kulcsnak helye nincs. Akkor talán a helyettesítésnél? „Végtelen" számú helyettesítés közül választunk? Megintcsak: nem! A DES-nek nyolc darab S-boxa, helyettesítési táblázata van, amelyek az S-P hálózati ábrának megfelelően kötött pozíción tanyáznak. Ezek közül sem lehet válogatni. A végén kisül, hogy a DES-hez nem is kell kulcs! És valóban. A DES elketyegne kulcs nélkül, csak ebben az esetben mindig ugyanazt csinálná, így nem lenne nehéz „feltörni".

Kell egy kulcs!

Sajnos bele kell rondítanunk a DES eddig levezetett gyönyörűen primitív képébe, mert kulcs nélkül olyan algoritmus lenne, ami ha nyilvánosságra kerül, már ki is dobhatjuk. Érdekes, hogy valahogy a fejlesztők is úgy viszonyultak a kulcskérdéshez, hogy szinte fájt nekik, s ez meg is látszik a dizájnon – de ez semmit sem von le a kulcs értékéből.

A szép és logikus modell érintetlenül hagyásával úgy tehető kulcsfüggővé a működés, ha valahol a ciklusok között „belemosunk" némi idegen, külső adatot is a folyamatba, vagy más szóval: „megfűszerezzük" *(hivatalosan saltnak, sózásnak nevezik)* az amúgy ízetlen algoritmust. S hogy ez ne tegye tönkre csodálatos, egyértelmű szubsztitúcióinkat és permutációinkat, olyan módon kell „beledolgoznunk" a „fűszert", hogy mégiscsak le lehessen vakarni róla, ha muszáj. Márpedig kibontáskor muszáj lesz, mert különben nem az eredeti adatokat kapjuk vissza, hanem egy fűszeres adattrutymót.

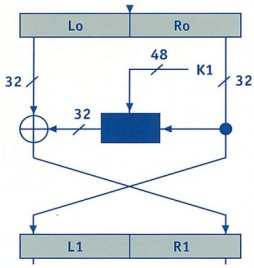
Vajon hogyan lehet egyik adat hátára úgy ráültetni egy másikat, hogy később le lehessen szedni onnan? XOR függvénnyel!

A DES bemeneti 64 bites kulcsából *(mely sajnos 8 paritásbitet tartalmaz, tehát 56 hasznos bitből áll)* 16 darab, egyenként 48 bites kiskulcs születik, mely a 16 körös szubsztitúció-permutáció közé ékelődve fokozatosan „beleivódik" az adatba. Kibontáskor, miközben visszafelé permutálunk és helyettesítgetünk ugyanezeket a kiskulcsokat „lexoroljuk" az adatról.

Itt valami nem stimmel. A 48 bites kiskulcsok mérete ugyanis nem egyezik meg a bemeneti 64 bites adathalmaz méretével!

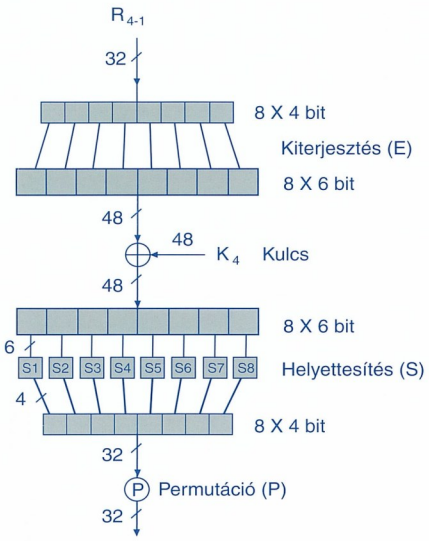
A DES egy lépése részletesen

A DES a bemenő adatblokkot kétszer 32 bitre bontja *(bal és jobb)*. Az egyes fordulókban a jobb blokk következő tartalma kemény munkával áll elő *(S és P, mind a bal, mind a jobb blokk adatai alapján)*, míg a bal blokk lustán felveszi a jobb blokk előző értékét. Az alábbi ábra egy DES ciklust mutat, ilyenből 16 követi egymást. L0 és R0 a bal és jobb blokk kiindulási állapota, L1 és R1 pedig – értelemszerűen – a forduló eredménye. Leolvasható, amint a jobb oldal *(R0)* átfut egy fekete dobozon *(ez a S-P lépés, maga a titkosítás)*, amit felülről egy 48 bites kiskulcs *(K2)* fűszerez. A dobozból kilépő adat összexorolódik az L0-val, és ez lesz a jobb oldal következő kiindulási állapota. L1 pedig egyszerűen R0-ból táplálkozik.



A 16 egyforma DES lépés menete. Itt a piros, hol a piros?

Jól jegyezzük meg a fenti sémát: ez Feistel találmánya *(IBM szabadalom: 1974 március 19.)*! A fekete doboz szabadon cserélhető az eljárásban. A Feistel-típusú blokktitkosítások mindegyike erre a blokkfelezős, itt a piros, hol a piros megoldásra épül, ezt hajtja végre ciklikusan. Már csak a fekete doboz felderítése van hátra. Vajon hogy fűszerezünk 32 bites adatokat 48 bites kiskulccsal? Hát úgy, hogy mielőtt XOR-ra kerülne a sor, a 32 bites adatot 48 bitesre pumpáljuk! Az alábbi ábra mutatja a fekete doboz belső működését:



A DES működése bitszinten

A változatosság kedvéért a 32 bitről 48 bitre pumpálást is egy kötött táblázat segítségével végzik. Az E *(expanzió)* táblázat is publikus, nem valami látványos, de így fest:

E					
32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

DES expanziós táblázat a 48 bites átalakításhoz

Miután az adat felfúvódott, hozzáxoroljuk a 48 bites kiskulcsot, majd a – szintén kötött táblázatos – S-boxok segítségével egyszerű helyettesítést végzünk rajta. Ebben a lépésben az adathossz visszacsökken 32 bitre. Ezután a P-táblázat segítségével permutálunk egy jóízűt. Érdekes, hogy bármit is kavartunk idáig, az mind reverzibilis, visszafordítható! Mind az S, P, mind az E táblázatok úgy vannak kialakítva, hogy a folyamat megfordítható legyen, hiszen a titkosítás megszüntetésekor ugyanezeken a lépéseken visszafelé is végig kell tudni menni.

Emlékező titkosítások

A blokktitkosítások mindegyike könnyebben törhető, ha a titkosított adatfolyam blokkjai között nincs összefüggés, minden blokk kibontása csakis önmagától és a kulcstól függ. A független blokkok előnyt jelentenek, ha az adatátviteli közeg zajos, sokszor hibázik, mert egy-egy blokk elvesztése nem teszi lehetetlenné a titokzatos adatok maradékának kibontását. Másképpen fogalmazva: az algoritmus nem „emlékszik" korábbi hibákra. Biztonsági szempontból azonban ez a megoldás kívánnivalókat hagy maga után. Egyrészt ez annyit jelentene, hogy ugyanazzal a kulccsal ugyanazt az ismétlődő mintát titkosítva a titkosított adathalmazon is felismerhetők lennének az ismétlődő minták. *(Ez volt az Enigma hibája, emlékezzünk, EIN-eket lehetett keresni benne, mivel minden karakter önállóan kódolódott. Az Enigmának ilyennek kellett lennie.)* Másrészt szinte tálcán kínálja a titkosított adathalmaz részeinek lecserélését, mert úgysem veszi észre senki. Ha az adatfolyam például egy banki átutalás, meg lehetne próbálni a kényes pontokon átfirkantani, hátha sikerül jó nagy kalamajkát okozni. Napjaink hálózatai vagy egyáltalán nem hibáznak, vagy ha igen, azt észrevétlenül, a háttérben korrigálja a TCP/IP. Ilyen megbízhatóságú közegben érdemes élni a blokkok összefűzésének lehetőségével, mert ugyan a hibák „emlékezetessé" válnak, de egyedi blokkokat többé nem lehet átírni a folyamban, és nincs értelme töredékeken kriptoanalízist folytatni. A blokktitkosításoknál leggyakrabban bevetett láncolási módszer a CBC *(Cipher Block Chaining, titkosított blokkok felfűzése)*.

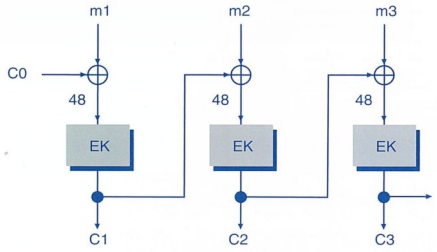
CBC

Egyszer már felmerült a kérdés: hogyan lehet egy adathalmaz hátára feltenni egy másik adathalmazt, hogy az később levakarható legyen? XOR-ral! Nem meglepő tehát, hogy a CBC az egyes DES blokkokat xorolással viszi rá a következőre, s ezzel egyfelől függőséget teremt a blokkok között, másfelől lehetetlenné teszi ismétlődő minták felismerését. Az alábbi ábra a CBC sematikus

vázlata, bal oldalon a titkosítással, jobbra a visszafejtéssel. Amint leolvasható, a folyamat során az éppen következő bemenő blokkot még a titkosítás előtt összexoroljuk az előző blokk titkosított változatával. Majd jöhet a Feistel-ciklus, melynek kimenete a következő inputblokkhoz xorolódik - és így tovább. A CBC beindításához kell egy nulladik blokk, ami a legelső inputblokkhoz xorolódik, ezt reprezentálja a C0 *(nulladik cipherblock)*. Kibontás:

1. a kulccsal kinyitjuk a legelső blokkot, ezután lexoroljuk a hátáról C0-t: kész az első kibontott blokk
2. a kulccsal kinyitjuk a második blokkot, és lexoroljuk róla a legelső blokk titkosított változatát
3. és így tovább.

C0 akármi lehet, titkosnak sem kell lennie, mert a DES kulcs nélkül úgysem lehet levakarni a legelső blokkról. E nélkül pedig a lánc sem fejthető vissza.



CBC láncolás a DES alatt

Ennyi volt a DES. Máshol fél évnyi tananyag, nálunk négy és fél újságoldal. Remélem nem volt túl nehéz. Egy-két dolgot elnagyoltam *(az S-boxot megválasztásának matekját, a kiskulcsok generálását)* de ezek már nem szükségesek a folyamat megértéséhez. Remélem most már elhiszik, amit sok-sok matematikus állít: a DES feltörésének egyetlen biztos módja a nyers erő. Vannak ugyan gyenge pontok az algoritmusban, de ezek többnyire gyenge kulcsokra vezethetők vissza. A DES-nek négy olyan kulcsa van, mely egyáltalán nem titkosít, s néhány tucat gyenge kulcsról is tudunk. A többi egyelőre ellenáll a matematikusoknak is. Apropó visszafejtés!

Kriptoanalízis IV. - redundanciaalapú megközelítés. Nyers erő, nyers erő, de mit keresünk? Csak próbálgatjuk sorban a DES kulcsokat, hátha az egyik nyitja az adathalmazt – de honnan tudjuk, hogy megvan, amit keresünk? Másodpercenként akár tízezer DES-nyitást is könnyű kivitelezni, de ki mondja meg a programnak, hogy mikor találta meg a valódi kulcsot? Itt két módszer kívánkozik:

- ha ismerjük a titkosított adatok szerkezetét *(pl. egy képfájl)*, utasíthatjuk a brute-force programot, hogy álljon meg, ha felismerhető képformátum lett a kibontás eredménye
- ha nem ismerjük a szerkezetét, keressünk benne rejtett redundanciát! Ha szöveget *(pl. html lapot)* titkosítottak, az írás karakterei között nem lelhetők fel bizonyos jelek, például a hexa 30 alattiak. Utasítsuk a brute-force programot, hogy álljon meg, ha olyan visszafejtést talált, amiben viszonylag kevés hexa 30 alatti karakter van: megvan a html doksi! Ennél még egyszerűbb a Base64 kódolású adatok brute-force megtalálása: ha a visszafejtett szöveg csak az angol ABC kis-és nagybetűit tartalmazza, nyertünk!

Csak azt nem árultam el, hol itt a redundancia. Ha elolvassák a Huffmann kódolásról szóló cikkemet, tudni fogják, hol bújik meg: a kihasználatlan bitkombinációkban!

Mivel a DES matematikailag stabil *(csak a kulcs rövid)*, nem meglepő, hogy sokan abban látják a biztos jövőt, ha ezt a veteránt kipofozzuk egy kicsit. Így született a DESX és a 3DES.

3DES, DESX

A 3DES *(ejtsd: tripla-DES)* egyszerűen három DES titkosítás egymás mögé állítása. Mint ahogy a sima szubsztitúciók sorozata erősebb titkosítást ad*(hat)*, a sima DES-ek egymásutánja is ilyen hatással jár. A hatékony kulcshossz 3*56 bitre nő, a brute-force ideje pedig kismilliószorosára. A 3DES úgynevezett EDE módban használja a benne lévő három DES-t, ahol EDE nem egy férfinév, hanem az Encryption-Decryption-Encrypt rövidítése. A három DES közül az első titkosítási, a második *(más kulccsal!)* visszafejtési, az utolsó ismét *(a harmadik kulccsal)* titkosítási irányban fut. A másodiktól nem kell félni: a visszafejtés itt egyszerűen a DES-CBC és a Feistel ciklus fordított használatát jelenti. A DES ugyanis - mint tudjuk – megfordítható.
A DESX szintén a kulcshossz növelésére született trükk. Az RSA Laboratories dolgozta ki, s lényege, hogy az 56 bites DES kulcson kívül további két, egyenként 64 bites kulcsot alkalmaz. Egyiket a blokk titkosítása előtt xorolják rá az adatra, másikat a titkosítás után, de még a CBC előtt. Ezzel a kulcs 56+64+64 bitesre nőtt. Bizonyos, itt ki nem fejtett *(lineáris, differenciális)* kriptoanalízis-módszerek előtt nyílik, mint a budiajtó, de ezekhez speciális tudás és mintaadat is kell, így mindennapi használatra 186 bitesnek tűnik.

Ennyit a DES-ről, a többi szimmetrikus algoritmus pedig egyelőre ráér. Van még egy-kettő, amit érdemes lehet megismerni – bár a Windows világban nem bukkannak fel: IDEA, RC2, RC4, RC5, FEAL, SAFER, LOKI, KHUFU, BLOWFISH, CAST, SHARK, BEAR, LION, SKIPJACK, 3WAY, Rijndael, SERPENT,

Fóti Marcell
MZ/X
marcellf@netacademia.net

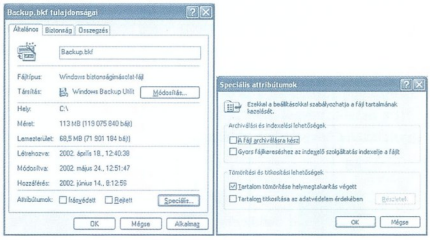
A cikkben szereplő URL-ek:

[1] Az Enigma-sztori http://ed-thelen.org/comp-hist/NSA-Enigma.html
[2] RSA FAQ http://www.rsasecurity.com/rsalabs/faq/index.html
[3] Applied cryptography, kinyomtatható PDF fejezetekkel http://www.cacr.math.uwaterloo.ca/hac

XP: tömörítések

A Windows XP összes változata *(Professional, Home)* NT alapú. Ha pedig NT, értelemszerűen a korszerű, modern NTFS fájlrendszert választjuk, nem a régi öreg, korlátolt FAT-et, vagy tuningolt változatát, a FAT32-t. Ezzel kismillió szolgáltatáshoz *(tranzakciónapló, jogosultsági rendszer, kvóta stb.)*, jutunk, és kezünkbe kerül az on-the fly *(röptében)* tömörítés is. Így már két tömörítési eljárás közül választhatunk, hisz az XP „zipelni" is tud. Mikor melyiket érdemes használni?

Mindenekelőtt szögezzük le: mind a röptében tömörítés, mind pedig a ZIP ugyanazt a veszteségmentes tömörítést, a Lempel-Ziv algoritmust használja, mely mögött az úgynevezett Huffmann kódolás áll. Először áttekintjük a tömörítések menetét. Azok kedvéért, akiknek az egerészés a könyökén jön ki, a cikk második felében az LZ algoritmussal foglalkozunk: puszta kézzel betömörítjük az „én elmentem a vásárba félpénzzel" stringet.
A két tömörítési módszer az azonos algoritmus ellenére is különböző mértékű tömörítést tesz lehetővé, ugyanis az LZ többféleképpen *(sebességre, vagy tömörítési arányra)* optimalizálható. A régi motorosok kedvéért: az ARJ parancssori változatában ez a tuningolás teljesen a kezünkben volt: akár a Huffmann puffer méretét is át lehetett állítani. Ma már az ilyen szintű hozzáférés nem divat.



A röptében-tömörítés párbeszédpaneljai

A röptében-tömörítés

Kezdjük az NTFS magánszámával, a Windows NT 3.51-ben megjelent áttetsző *(opaque)* tömörítéssel. Miért nevezik áttetszőnek? Mert a tömörítés a háttérben, a felhasználói folyamatok zavarása nélkül zajlik le: a tömörített fájlok zavartalanul használhatók, mivel az operációs rendszer minden megnyitáskor a memóriában kibontja, minden mentéskor röptében tömöríti, és zsugorformában teszi a lemezre.
Olyannyira átlátszó a folyamat, hogy ha az Explorert nem utasítjuk, hogy a tömörített fájlokat más színnel jelezze, egyszerűen fel sem tűnik, hogy mi tömör, és mi nem. *(NTFS-szinten a tömörített állapot jelzésére egy fájlattribútum szolgál, mely a Hidden-System-Read only-Archive négyesfogat ötödik lova.)*
A tömörítés menete: tetszőleges fájlon vagy könyvtáron vegyük elő a tulajdonságlapot *(amelyen a modern idők szellemében csak két attribútumot látunk az imént felsorolt öt közül)*, kattintsunk a „Speciális..." gombra, s a megjelenő „Speciális attribútumok" ablakban ízlésesen helyezzünk el egy pipát a „Tartalom tömörítése..." jelölőnégyzetben. A megfelelő számú oké után az XP nekiesik a fájlnak, és tartalmát mintegy ötven százalékos arányban tömöríti. *(Ez fájltípustól, méginkább a fájlon belüli redundancia mértékétől függ. Végrehajtható fájlok (DLL, EXE) esetében tipikusan ötven százalék körüli tömörítési arány várható.)*
Az alábbi párbeszédpanel-páros az imént említett pipákon kívül egy 130 megabájtos backup fájl tömörítési arányát is mutatja. Az NTFS fájlrendszerben a fájlokról tárolt adatok mennyisége sokszorosa a FAT-en megszokottnak, így nem meglepő, hogy adatállományainknak két mérete is van: egy valódi, és egy tömörítés utáni *(az ábrán „Lemezterület" néven szerepel és 65,8 MB-ot mutat)*.

Ha belegondolunk, ez a tömörítési forma még arra is jó lehet*(ne)*, hogy kvótával ellátott köteteken helyet szabadítsunk fel saját magunk számára. A gondolat kiváló, de sajnos tett nem követte: a redmondi fiúk valamilyen speciális okból a kvóta kiszámításához nem a ténylegesen elfoglalt „Lemezterület", hanem a becsomagolás előtti „Méret" mezőt használták fel. Hogy miért? Csak.

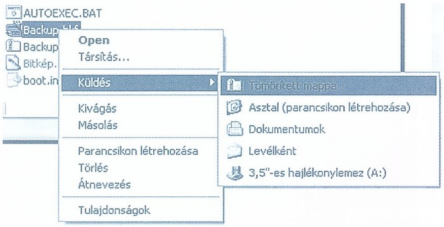
A röptömörítés és a hálózat

Gondolatkísérlet: ha két XP hálózaton kommunikál egymással, s az egyik gépről a másikra átkérünk egy röptömörített fájlt, nyilván tömörítve megy át a hálózaton, mivel az algoritmust mindkét fél ismeri. Teljesen felesleges a feladó oldalán kicsomagolni ugye? Igen. De nem ez történik. Network Monitorral kimérve sajnos jól látszik, hogy a fájl eredeti formájában halad át a hálózaton. Vajon miért? Mert a Windows fájlátviteli protokollja, a Server Message Block *(SMB)* a mai napig nem ismeri ezt a „tájszólást". Korábbi NetMon cikkeimben már mutattam, hogy ezerféle nyelven beszél *(Xenix stb.)*, hatszázféle fájlkezelési trükköt ismer *(hosszú fájlnevek, UNICODE stb.)*, de a tömörítésről még mindig nem tud. *(Az NT 3.51 megjelenése emlékeim szerint 1994-re tehető. Nyolc év nem volt elég a tömörítés általános elfogadásához...)*
Ha már a hálózaton „kitekerve" utazik, vajon a helyi gyorsítótárban legalább tömörítve marad? Nos, egynél több hálózati felhasználó esetén már mérlegelni kell, hogy a gyorsítótár *(cache)* használata mikor hatékonyabb: ha abban tömörített, illetve ha eredeti méretű fájlokat tárolunk? Paradox módon az eredetik tárolása hatékonyabb, mert bár több memória fogy, de ha sok felhasználónak kell eljuttatni ugyanazt az adatot, nem mindegy, hogy egyszer bontjuk ki, vagy állandóan ki-be csukogatjuk. A RAM olcsó, a processzorcsere kellemetlen: fogyjon inkább a RAM.

Az XP beépített ZIP kezelése

A Windows XP egyik nagy újdonsága, hogy immár felismeri és kezeli napjaink legeltertedtebb tömörített fájlformátumát, a ZIP-et. Nyitja, csukja, készíti – mindent tud, amire egy átlagos felhaszná-

lónak szüksége van. Mostantól csak a trükkös felhasználók számára van értelme WinZIP-et és hasonló célszoftvereket használni, a mindennapi tömörítések egyetlen kattintással elvégezhetők. Így: álljunk rá a tömörítendő fájlokra, és a jobbkllikkes menüből válasszuk a Küldés->Tömörített mappa menüpontot. Az elnevezés ne zavarjon meg senkit: nem mappa, hanem hagyományos ZIP fájl születik!



A ZIP-tömörítés legegyszerűbb módja

Mennyiben más az így kapott, zipzáras „mappában" üldögélő tömörített fájl, mint a röptében tömörített? Először is lényegesen lassabban, de lényegesen kisebb fájlméretet kapunk. Az előző 130 megabájtos mentési fájl ZIP-mérete mindössze 45 megabájt! Másodszor a ZIP-tömörítés nem átlátszó. Ha ebből a „mappából" megnyitunk egy fájlt, az suttyomban kicsomagolódik a profilbéli temp-könyvtárunkba, s így nyílik meg. Szó sincs helyben, röptében tömörítésről! Kicsit kényelmetlenebb ugyan használni, de a mai napig ez a fájlformátum az, mely garantáltan tömörítve utazik a hálózaton is!

Tömörítési algoritmusok

A redundancia kivonására alapvetően kétféle „erősségű" megoldás áll rendelkezésünkre: veszteségmentes „kivonat", melyből az adatok eredeti állapota állítható vissza, illetve a veszteséges, melyből az eredetire hasonló, de azzal minőségben meg nem egyező adatok nyerhetők vissza. Míg a veszteségmentes módszer minőségromlást nem okoz, éppen ezért kisebb arányú *(50-80%-os)* tömörítést tesz lehetővé, addig a veszteségesnél a veszteség növelésével *(és a minőség romlásával)* egyre nagyobb tömörítési arány érhető el *(80-99%)*. A veszteséges tömörítést általában multimédiás adatok méretének csökkentésére használják *(JPEG, MPEG, fraktál stb.)*. A minőségromlást nem okozó tömörítési módszereknek pedig például dokumentumok, programok tárolásakor és utaztatásakor vesszük hasznát *(Zip, NTFS)*.

Veszteségmentes tömörítés

Az egyik legprimitívebb eljárás az ismétlődések kiváltása. Remek tömörítési arányt lehet elérni DBase fájlokon pusztán azáltal, hogy a fix hosszúságú adatbázismezők töltelékszóközeit darabszám szerint összevonogatjuk, s két bájton leírjuk: itt 23 szóköz következik. Ugyanez az eljárás képfájlokkal is csodát művelhet. De mi a helyzet a folyamatos és változatos szövegekkel, illetve a ránézésre tökéletesen véletlenszerű, ismétlődésmentes bájtsorozatokból felépülő programokkal *(*.EXE, *.DLL)* ?
Ha egy textfájl tartalma „Karácsonyi üdvözlet", és csak ez a két szó, vagyis 19 karakter alkotja a tömörítendő adathalmazt, mindenki tudja, hogy nem érdemes tömöríteni: a „tömör" fájl nagyobb lesz, mint az eredeti. Miért? Vajon nincs a példaszövegben redundancia? Első ránézésre nincs, hisz nincs betűismétlődés sem. Azonban az első ránézés félrevezető. A redundancia ott bújik meg, hogy ezt a rövidke szöveget 8 bites karakterekkel, bájtokkal írtuk le, mely 256 különböző jel megkülönböztetését tenné lehetővé - de nekünk itt a szóköz beszámításával is csak 19 jelre lenne szükségünk, azaz 5 bites kódtábla is elegendő lenne. 19*8 helyett 19*5 bit tárolása igen jelentős tömörítési arány - lenne! A bökkenő az, hogy míg a 8 bites kódtáblát szabványosították *(ASCII)*, így minden gép ismeri, addig a mi 5 bites táblánk értelmezéséhez a tömörített adatok mellé le kell tárolnunk a visszafejtéshez szükséges ABC-t is, s máris 30-40 bájtnál tartunk az eredeti 19 helyett.
Ha továbbgondoljuk a rövidített ABC-eljárást, rájövünk, hogy hosszabb szövegek esetén ezzel nem lehet célt érni: ahogy a szövegben egyre több betű fordul elő, úgy illan el annak az esélye, hogy 8 bitesnél kisebb kódszótárral leírható a szöveg.

Gumibájt

Szerencsére épp idejében született Morse *(Samuel F.B. Morse 1791 – 1872, a Morse ABC szabadalmaztatása: 1840-ben)*, hogy zsenialitásának fényével megvilágítsa homályban tapogatódzó kezeinket. Mindenki ismeri a Morse ABC néhány betűjét, például az „S" betűt *(...)*, az „o" betűt *(---)*, vagyis el tudja olvasni ezt: ...---... A katonaviseltek még ennél is többet tudnak, például ismerhetik az "E" betűt, ami egyetlen rövid jelből áll csupán. A Morse ABC különlegessége abban áll, hogy a karakterkódok tárolásához változó számú "bitet" használ fel, így a természetes nyelvekben gyakoribb hangok kevesebb tárolóhelyet *(akár egyetlen bitet!)* emésztenek fel, míg a ritkábban előfordulók akár négy-öt bitet is igényelhetnek. Ha az egyszerű szöveges fájlokat nem ASCII, hanem Morse kódolással tárolnánk, 80-90%-os tömörítési arányt érhetnénk el! Kolosszális, zseniális! A III. évezred ötlete!

Akkor miért nem így tároljuk...?

1. Mert a Morse ABC nem bináris, hanem háromállapotú jelekből épül fel *(hisz külön értelmezendő a jel hiánya, a szünet is)*, s ezt igen nehéz lenne átültetni az ostoba, betokosodott bináris számítógépeink nyelvére.
2. A Morse ABC gyakoriságfüggvénye kizárólag nyers dokumentumokra használható, de ha a fáljban egyéb bináris információ is szerepel ***(ilyen például a Word .DOC dokumentumok fejléce)***, az angol nyelv statisztikai elemzésén alapuló gyakoriság tökéletesen használhatatlanná válik. Nem is beszélve a szövegnek igen nehezen nevezhető végrehajtható programokról.

A változó hosszúságú kódolás ötletét azonban ne vessük el, hisz zseniális. Ha a kódtáblát mindig az adott tömörítendő adattartalom alapján egyedileg, dinamikusan építenénk fel, tetszőleges bináris folyam tömöríthetővé válna. S ezzel feltaláltuk a Huffmann kódolást. A világ összes veszteségmentes tömörítője ezen az elven, vagy az ebből kifejlesztett Lempel-Ziv algoritmus alapján működik *(ARJ, ZIP, PK, LHA stb.)*. A tömörítőprogram első lépésként felállítja a tömörítendő fájl bájtjainak gyakoriságtáblázatát, majd ennek birtokában „leosztja a lapokat", vagy egy jobb hasonlattal élve szótárt készít. A gyakoribb bájtokhoz rövidebb *(2-3 bites)*, a ritkábban előfordulókhoz hosszabb *(akár 8 bitet is meghaladó)* kódot rendel, majd e szótár segítségével "lefordítja" a fájlt, s végül belekerül a „szótár" is, hogy a kicsomagoláskor ne kelljen külön szótárfájllal vesződnünk. *(Aki elég régi motoros a szakmában, találkozhatott olyan őstömörítővel, melyből két fájl pottyant ki: egy tömörített eredmény meg egy szótár.)*
Vajon miben különbözik egyik tömörítő a másiktól? Sebességben, tömörítési arányban, és szolgáltatásválasztékban. De az algoritmusuk ugyanaz: Huffmann vagy Lempel-Ziv. A teljesítménykülönbség, és az egymással nem kompatibilis fáljfelépítés

abból adódik, hogy mindegyik gyártó maga próbálja meg megtalálni az optimális kódszótárt, amely hite szerint jobb, mint a konkurenciáé. A szótár felépítésére nincs szabály, a lényeg, hogy hatékony tömörítést tegyen lehetővé. Minden gyártónak megvan a maga féltve őrzött titkos receptje, s legalább annyira titkolják egymás elől mint Zwack az Unicumot.

A Huffmann kódolás

Írjunk veszteségmentes tömörítőprogramot! Nem kell hozzá más, mint papír és ceruza. Legyen a feladat az „én elmentem a vásárba fél pénzzel" nótasor tömörítése.

1. lépés: gyakoriságtáblázat készítése

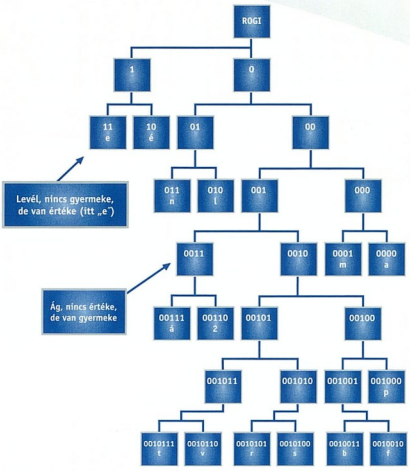
Elsőként meg kell számolnunk, melyik karakter hányszor fordul elő a szövegben. Az alábbi táblázat ennek a bonyolult számításnak az eredményét tartalmazza:

Előfordulási gyakoriság	Karakterek
egyszer fordul elő	t, v, s, r, b, f, p
kétszer fordul elő	m, a, á, z
háromszor fordul elő	é, n, l
négyszer fordul elő	e

A táblázat alapján a legrövidebb kódsorral az "e" írandó le, mert ebből van a legtöbb. *(Valójában a leggyakoribb betű a szóköz (5 db), de ezt teljesen kihagytam a további számításokból, mert nehéz lerajzolni, így valójában ezt tömörítjük: énelmentemavásárbafélpénzzel)*

2. lépés: kódszótár felépítése

A kódszótár kialakítása döntő fontosságú, a tömörítés hatékonysága a kódszótár helyes megválasztásán áll vagy bukik. Nincs szabványos, vagy egyedül üdvözítő algoritmus a szótár összeállítására, csak ökölszabályok, irányelvek léteznek. Ezek közül talán a legfontosabb, hogy a kódrendszer legyen „önjáró", egyértelmű. A Morse ABC például nem lenne egyértelmű, ha megfosztanánk a harmadik „dimenziójától", az időtől *(vagyis a szünetjelektől)*, mert vannak olyan elemei a szótárnak, melyek más elemek építőköveiként megtalálhatók. A „..." jelsorozat jelenthet egyetlen „S" betűt, de akár így is olvashatjuk: „EEE". A mi kódszótárunk nem tartalmazhat ilyen ellentmondást! Az egyik legjobb módszer egyértelmű szótár kialakítására a bináris fára felaggatott ABC, ahol csak a fa levelei hordozhatnak szótárelemeket, az elágazások nem. Képzeljünk el egy bináris karácsonyfát, melyet az ABC betűivel díszítünk. A fa csúcsának közelébe érdemes felakasztani az értékesebb díszeket, tehát a gyakori betűket *(hogy ne érjék el a gyerekek :)*, míg lejjebb elhelyezhetjük a kevésbé értékeseket. A fa lehet magas is és széles is, az alábbi ábrán magas fát „díszítettem", amelynek előnye, hogy a leggyakoribb jel *(a mi esetünkben az E)* két bites jelsorozatot kap a szótárban - de ez egyben a hátránya is: a ritkább jelek rengeteg bitből állnak.



Tömörítési kódszótár építése

Szélesebb, de alacsonyabb fát is készíthettem volna azáltal, hogy nem helyezek mindjárt a csúcs közelébe jelet, hanem előbb hagyom egy kicsit „bokrosodni", s a rengeteg ágra aztán 1-2 réteg alatt felaggatható a sok szép betű.

A fáról már könnyen leolvasható az egyes betűknek megfeleltetendő bitsorozat, amelyet az alábbi táblázatban gyűjtöttem össze:

e	11	p	001000
é	10	t	0010111
n	011	v	0010110
l	010	r	0010101
m	0001	s	0010100
a	0000	b	0010011
á	00111	f	0010010
z	00110		

Ha valakinek van kedve hozzá, végigbogarászhatja, hogy ABC-nk egyértelmű-e, vagyis található-e benne olyan önellentmondás, mint a Morse ABC-ben *(Emlékeztetőül: S=EEE)*

3. lépés: átkódolás

A kódszótár birtokában elvégezhetjük a „fordítást", s lemérhetjük tömörítésünk hatékonyságát

énelmentemavásárbafélpénzzel *(28 bájt)*

=10011110100001110110010111110001000000101010011100101000011100101010010011000000100101001000010001100111001101011010 *(14 és fél bájt)*

Ez körülbelül 50%-os tömörítési aránynak felel meg. Nem rossz, de nem is jó, aminek az az oka, hogy túl magas lett a fám, ezért aránytalanul sok karakter kapott 7 bites kódot. Lelkes Kedves Olvasóim elkészíthetik saját „díszítésüket", hogy lemérjék, vajon kódszótáruk hatékonyabb lesz-e *(ebben egészen biztos vagyok)*, mint az én magas fám.

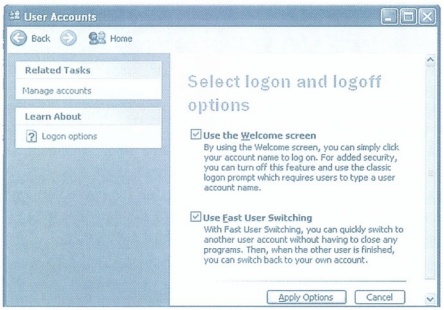
Fóti Marcell
marcellf@netacademia.net
MZ/X

A Windows XP zavaró újdonságai

Emlékszem, amikor először használtam Windows 2000-et, a felhasználói felületen megjelenő újdonságok, a sok varázslat, ami ugrásra – akarom mondani kattintásra – készen állt, hogy igényeimet kielégítse, rendkívül idegesített. Minden akkori tudásomat latba vetve majdnem az első dolgom volt, hogy valami használhatóbb, kevesebb extrával felszerelt környezetet teremtsek magamnak, ahol mindent látok, amit kell, könnyen megtalálom a programokat, kicsit otthonosan mozgok a sok újdonság közt. Nem telt el sok idő, hozzászoktam az általam beállított, új környezethez. Aztán jött a Windows XP - hamarabb, mint gondoltam. Újból felborította a rendet, bár közel sem annyira, mint elődje a Windows 2000. Persze itt is a megújult felhasználói felület - alig lehet valamit ott megtalálni, ahol megszoktuk – a leginkább zavaró. Mindez állítólag a felhasználók, vagyis a még könnyebb és hatékonyabb munkavégzés érdekében történik.
A Windows XP-re jellemző a minden eddiginél több pop-up ablak és varázsló, új ficsörök sokasága, amelyeket valójában kevés helyen használnak. Meglepődve tapasztaltam, hogy sok rendszergazdát inkább az érdekel, hogyan is tüntesse el az újdonságokat a felhasználók elől, mert tulajdonképp nincs szükség rájuk. Összeszedtem hát egy csokorba őket, mindenki okulására.

CTRL+ALT+DEL bejelentkezés

Kezdjük mindjárt a belépőképernyővel. Egészen addig, míg a gép nem tartomány tagja, lehetőségünk van egyszerűbb módon belépni, mint az eddigi CTRL+ALT+DEL-es módszerrel. Amint tartomány tagjává tesszük a gépet, visszatér a hagyományos bejelentkezés.
Ahhoz, hogy tartomány nélküli környezetben visszakapjuk a már ismert bejelentkezési képernyőt, a Control Panel – User Accounts beállítások közt az alábbi képen is látható két pipát kell megszüntetni:



A CTRL+ALT+DEL „visszaszerzése"

A Fast User Switching kikapcsolása csupán azt biztosítja, hogy minden esetben ki kell lépni, mielőtt valaki más beléphetne a gépre, ettől még a bejelentkezési képernyő nem fog változni, viszont Welcome Screen nélkül sajnos nincs lehetőség gyors felhasználócserére sem. Az igazán hasznos gyors felhasználóváltás tehát a kiskacsás bejelentkezőképernyő függvénye – az ördög tudja miért! Ebből sajnos az is következik, hogy tartományi tag gépeken *(ahol CTRL+ALT+DEL-es a bejelentkezés)* a gyors felhasználóváltás nem elérhető.

Windows XP Tour

Az első három belépés alkalmával a Windows XP kis kitérőre invitálja a felhasználót, mert szeretné bemutatni önmagát.

Az XP ezzel a buborékkal hívja fel magára a figyelmet

Mivel csak az első három belépésnél jelenik meg az invitálás, ezért ezt a problémát az idő is megoldja, de a registry módosításával elkerülhető a szívélyes invitálás.
Az új felhasználókra vonatkozóan a HKEY_LOCAL_MACHINE alatt kell változtatni, a többiek esetében a HKEY_CURRENT_USER alá kell felvenni a következő értéket:

```
[HKEY_LOCAL_MACHINE vagy HKEY_CURRENT_USER]
\Software\Microsoft\Windows\CurrentVersion\Applets
\Tour
Value: RunCount
Type: REG_DWORD
Data: 0
```

Ha az utolsó kulcsot nem találjuk, magunknak kell felvenni azt a regisztrációs adatbázisba.
Ha mégis szeretnénk végigjárni a bemutatást, a Start Menu – Accessories alatt továbbra is megtalálható a Windows XP Tour ikon, bármikor elindítható.

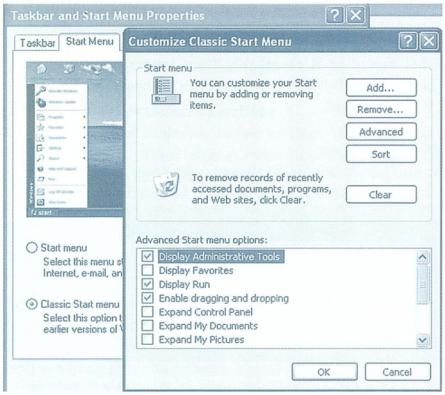
Az ablakok és a gombok

Nemcsak a Start Menü, hanem az ablakok és gombok arca sem tetszik sokaknak. A Control Panel – Display – Appearance lapon vissza tudunk térni a Windows Classic stílushoz. Sok választási lehetőségünk nincs, a Windows XP-s és a Windows Classic változat közt lehet dönteni. Tapasztalataim szerint a legtöbben a Classic változatot választják.
A rendszergazdák házirendben is meghatározhatják az ablakok megjelenését. Az erre vonatkozó beállítások a házirend felhasználói beállításai közt az Administrative Templates\Control Panel \Display\Desktop Themes alatt találhatók.

A Start Menü

Elég sokan nem szeretik a Windows XP új Start Menüjét, ezért inkább szeretnék visszakapni a régit. Érdekes megközelítés az

üres Asztal is, hisz Bill Gates annak idején ezt mondta: „Számítógépet minden asztalra!" S most tessék, az XP asztalán nincs semmilyen számítógép. Furcsamód a teli, terülj-terülj asztalkám és a hagyományos Start menü beállítása összenőtt, a klasszikus Start menü kiválasztásával klasszikus asztal is jár - az ördög tudja miért! A megoldást mindenki meg is találja, a Taskbar – jobb klikk – Properties ablakig remélem mindenkinek sikerül eljutnia.



Visszatérés a régi Start Menühöz

Itt egyszerűen visszakapcsolhatunk Classic Start menüre, vagy tovább mehetünk és a „Customize..." gomb mögött, további finomításokra van mód.
A házirendben meglehetősen aprólékosan lehet szabályozni a Start Menü és Taskbar beállításait. A felhasználói beállítások közt negyvenkettő különböző állítási lehetőség található az Administrative Templates \ Start Menu and Taskbar alatt. Jó bogarászást!

Windows Messenger

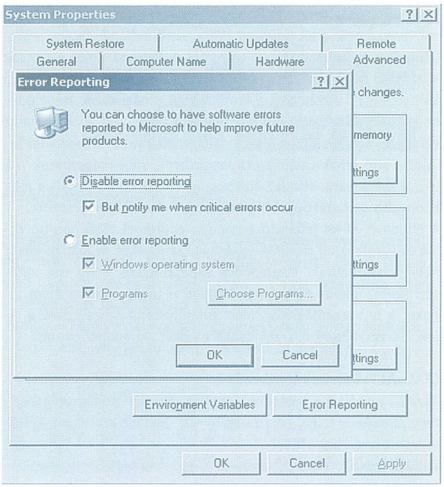
A Messenger letiltásának többféle módja ismeretes. Én itt most csupán egyet ismertetek, a Microsoft Knowledge Base Q302089 cikk tartalmazza többféle Messenger verzió esetén a támogatott megoldásokat.
Az egyik lehetséges megoldás, ha a házirend segítségével tiltjuk le a Messengert. Két helyen is meg lehet találni, egyszer a felhasználók, egyszer pedig a gép beállításai közt. Mind a két helyen a az Administrative Templates \ Windows Components \ Windows Messenger útvonalon van a beállítás.

Messenger letiltása a házirendben

Az elsővel abszolút megtiltjuk, a Windows Messenger futását, míg a második arra jó, hogy bejelentkezéskor ne indulhasson el a Messenger.
Ha mind a kettőt „Enabled" állapotra hozzuk, nem tudják majd használni a felhasználók a Messengert, és a gép indulásakor sem töltődik be. Ha a gép és a felhasználók beállításai közt is átállítjuk, a gépre érvényes házirend fog érvényre jutni.

Hibabejelentés

Amikor egy alkalmazás vagy maga az operációs rendszer szabálytalan műveletet végez, vagy egyéb hibák miatt „elszáll", a Windows XP feldob egy ablakot, melynek segítségével el lehetne küldeni a Microsofthoz a hibát – feltéve, hogy a gépnek van Internetkapcsolata. Ha nem szeretnénk, hogy ez az ablak megjelenjen, a Control Panel – System ablak Advanced lapján, az Error Reporting gomb mögött rejtőző beállításokkal le lehet tiltani a hibabejelentést. Gondoljunk szegény felhasználóra! Nem is fogja érteni, hogy mit, hova, miért kellene neki küldeni, az elveszett munkája miatt aggódik inkább. Ez a szolgáltatás adja a legjobb táptalajt az amúgy is burjánzó legendakörnek *(keress rá a neten: urban legends)*, mely egységesen arról szól, hogy gépünk a Microsoft beépített ügynöke, kémje, és minden adatunkat jelenti Billnek.



A hibabejelentés kikapcsolása

Talán érdemes bekapcsolva hagyni a „But notify me when critical errors occur" funkciót. Így kritikus hiba estén az operációs rendszer értesíti a felhasználót az eseményről. Ha nem is tud vele mit kezdeni, legalább tud valamit mondani a rendszergazdának a történtekről.
Ha nem szeretnénk teljesen letiltani a hibabejelentést, a „Choose Programs" gomb alatt pontosan beállítható, hogy mely alkalmazások esetén próbáljon az operációs rendszer riportot küldeni.
A hibabejelentést szabályozhatjuk házirend segítségével is. A számítógépbeállítsok alatt a Administrative Templates\System\ Error Reporting alatt ugyanaz beállítható mindenkire egységesen, mint amit a System tulajdonságai közt találunk.

Beépített ZIP tömörítő

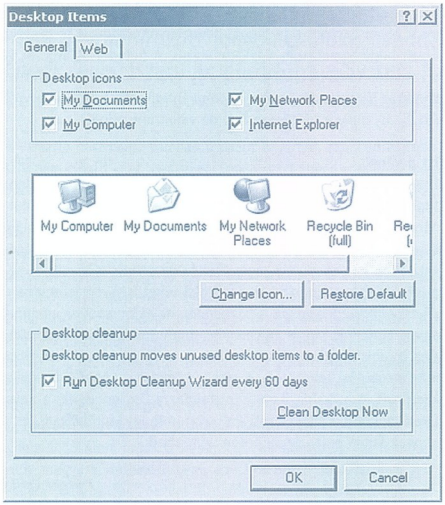
A beépített tömörítést a system32-ben található zipfldr.dll végzi. Előfordulhat, hogy a beépített tömörítő összeakad valamely külső tömörítő eszközzel, ezért hasznos lehet, ha tudjuk, hogy is kell kikapcsolni. A kiiktatásához a következő parancsot kell kiadni:

```
regsvr32 /u %systemroot%\system32\zipfldr.dll
```

A gép újraindítása után már nem fog működni a beépített tömörítő. Ha később mégis használni kellene, akkor a /u nélkül kiadva ugyanezt a parancsot, újra használhatóvá válik a funkció. *(Lásd még ZIP cikkünket a 22. oldalon!)*

„Desktop cleanup" varázsló

Alapértelmezett beállításként minden második hónapban elindul az asztaltakarító varázsló, amellyel a nem használt ikonokat távolíthatjuk el az asztalról. Abban ugyan van valami, hogy amit nem használunk két hónapig arra nincs is szükség, de mégis egyszerűbb kikapcsolni.



Az asztaltakarító varázsló kiiktatása

Ehhez a Control Panel – Display - Desktop ablakán a Costumize Desktop gombra kell bökni, ekkor jelenik meg a fenti képen is látható ablak. Itt kikapcsolhatjuk, vagy akár azonnal futtathatjuk is a varázslót, ahogy tetszik.

Szabályozhatjuk ezt a beállítást a házirendben is. A felhasználói beállítások közt az Administrative Templates\Windows Components\Desktop útvonalon található a „Remove the Desktop Cleanup Wizard" házirend, amelyet ha bekapcsolunk, nem fog futni hatvannaponta a varázsló *(sőt, egyáltalán nem lesz használható)*.

„Low Disk Space" figyelmeztetések

Amikor az operációs rendszer úgy dönt, hogy kevés hely van valamely partíción - konkrétan 200 MB szabad hely alatt – figyelmeztet, hogy nincs elég hely. Ahogy csökken a szabad terület, egyre gyakrabban figyelmeztet, pedig egy értesítésből is tudjuk: kevés a hely! Persze a legjobb, ha úgy szüntetjük meg ezeket az üzeneteket, hogy valóban takarítunk a merevlemezen, de lehetőség van az üzenetek megjelenésének megakadályozására is. Ehhez a registryt kell módosítani.

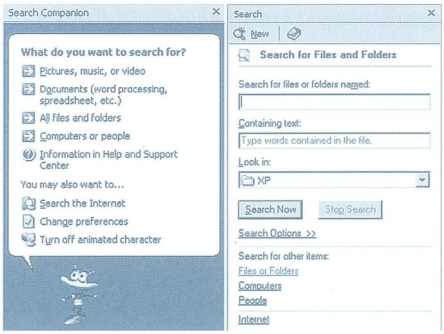
```
[HKEY_CURRENT_USER\Software\Microsoft\Windows\
CurrentVersion\Policies\Explorer]
Value: NoLowDiskSpaceChecks
Type: REG_DWORD
Data: 1
```

(Zárójelben jegyzem meg, ez a beállítás a registryben ugyanott található, mint azok, amelyeket a házirendben állíthatunk, én mégsem találtam a házirend szabályai közt. A Microsoft Knowledge Base is a registry módosítást ajánlotta (Q285107) erre a problémára.)

Figyelem! Érdemes legalább 200 MB szabad helyet tartani a partíciókon, mert ez szükséges a Windows XP System Restore funkció használatához. A fenti megoldás természetesen csak az üzenetek megszüntetésére jó, nem a probléma megoldására!

A keresési felület

Szintén egy olyan újdonság, amiről a legtöbben azt mondták, lelassítja a munkát. A Windows XP úgynevezett Search Companion újdonságáról van szó, amely a kereséseket hivatott megkönnyíteni.



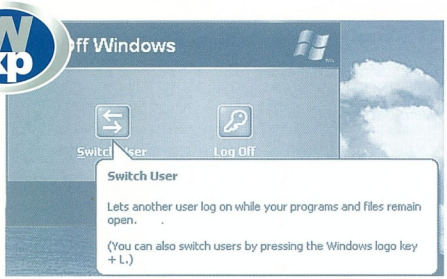
Keresési felület a Search Companionnal és nélküle

A visszatérés a Windows 2000-es keresési felülethez csak registrymódosítással lehetséges. A következő kulcsra van szükség:

```
[HKEY_CURRENT_USER\Software\Microsoft\Windows\
CurrentVersion\Explorer\CabinetState]
Value: Use Search Asst
Type: REG_SZ
Data: no
```

Balloon Tips

Mikor egy ikon fölé visszük az egeret és várunk egy másodpercet a kattintás előtt, a képen is látható úgynevezett „Balloon tip" – hívjuk buboréknak - jelenik meg, amiből megtudhatjuk, mi is fog történni a kattintás után. Ezen kívül a Taskbar vagy tálca jobb szélén levő értesítési területen *(Notification Area)* önmaguktól *(mondjuk belépés után, vagy egy-egy esemény következményeként)* további buborékok jelenhetnek meg, tájékoztatva bennünket a történtekről, vagy a szükséges tennivalókról. Ilyen tipp például a cikk második képén is látható Windows XP Tour figyelemfelhívó tippje is.



Egy buborék a sok közül

Megintcsak a regisztrációs adatbázis módosításával lehet kikapcsolni a buborékok megjelenítését:

```
[HKEY_CURRENT_USER\Software\Microsoft\Windows\
CurrentVersion\Explorer\Advanced]
Value: EnableBalloonTips
Type: REG_DWORD
Data: 0
```

Következő belépés után már nem fognak feljönni a tippek. Ne keverjük össze ezeket a tippeket a Start Menü ikonjain, vagy a fájlkezelőben a könyvtárak és fájlok esetén megjelenő információkkal. Ez utóbbi „üzenetet" az Explorer Tools menü - Folder Options – View ablakában tudjuk megszüntetni. Az Advanced beállítások közt alulról a második – a Show pop-up description for folder and desktop items – opcióval tudjuk szabályozni az infok megjelenését.

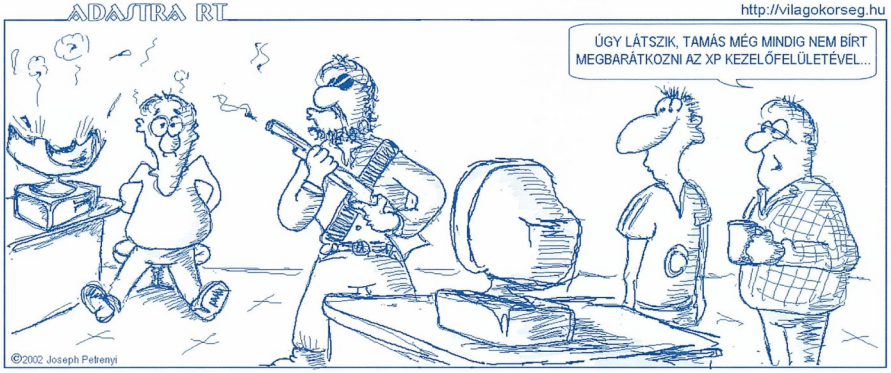
Windows Update

Automatic Update-nek is hívják, a menüben, mint Windows Update jelenik meg. Ez sem teljesen új funkció a Windows XP-ben, hanem a korábbi operációs rendszerekben *(Windows 95/98, Windows 2000)* fellelhető Critical Update Notification utódja. Nagyon hasonlít a Windows Millenium Automatic Update funkciójához.
A szolgáltatás időről időre megpróbál csatlakozni a Windows Update site-hoz, hogy letöltse a kritikus frissítéseket. Mindezt akár beavatkozás nélkül is képes elvégezni.
Lapunk hetedik oldalán kimerítő információ olvasható nemcsak a Windows Update-ről, hanem ennek vállalati hálózatra szánt kiszolgálókomponenséről, a Windows Update Corporate Editionról. S mivel annyi energiát szántunk arra, hogy megmagyarázzuk, mitől is olyan hasznos ez a bigyó, nem zárnám rövidre a történetét azzal, hogy bemutatom a kikapcsolását. A hetedik oldalon ez is olvasható, továbbá szóba kerül egy érdekes, ámde ismeretlen Windows szolgáltatás, a Background Intelligent Transfer Service *(BITS)* is, mely a háttérben fut, de vajon mit csinál?

Itt a vége...

Ebben a cikkben igyekeztem összeszedni a legtöbb olyan újdonságot, amelyektől – tapasztalatok szerint – szívesen szabadulnának az emberek. Ahol lehetett, megpróbáltam a felhasználó és a rendszergazda szemszögéből is megmutatni a megoldásokat. Több helyen csak a regisztrációs adatbázis módosítása segített, hiába van olyan sok beállítási lehetőség a házirendben. *(Persze írhatunk magunk is házirendet, de ebbe most ne menjünk bele.)* Mindenki vegye figyelembe, hogy Microsoft-ajánlás szerint, a registryt közvetlenül csak akkor módosítsuk, ha már nincs más megoldási lehetőség. A cikkben leírtakat többé-kevésbé a Microsoft Knowledge Base is tartalmazza, de mindenki a saját felelősségére használja őket. Én ugyan mindet kipróbáltam, és a gépnek kutya baja, de az ördög nem alszik.

Dorner Csilla
dorner.csilla@netacademia.net



Microsoft Exchange 2000:
Rendszergazdai jogosultságok

Csakúgy, mint az Active Directoryban, az Exchange-ben is van lehetőség arra, hogy meghatározott jogosultságokkal rendelkező kisrendszergazdákat nevezzünk ki. Ennek legegyszerűbb módja, ha a System Managerben az Exchange Administration Delegation Wizardot, vagyis a jogosultságosztó varázslót használjuk.

A varázsló használata valóban roppant egyszerű, azonban legtöbbször nem elégíti ki az igényeket. Éppen ezért ebben a cikkben megnézzük azt is, hogyan lehet finomabban hangolni a jogosultságokat.

Az Exchange jogosultságokról

Egy újabb bizonyíték az Active Directory és az Exchange 2000 összefonódására, hogy az Exchange a Windows 2000 jogosultsági rendszerét használja. Exchange telepítése során az Active Directory séma kibővül az Exchange specifikus jogokkal. Azt mondhatjuk tehát, hogy amikor a System Managerben az objektumok tulajdonságai közt átállítjuk a jogosultságokat, tulajdonképp Active Directory objektumok jogait változtatjuk.

Azokat a jogosultságokat, amelyek az Exchange nélkül is részei az Active Directorynak, Standard jogosultságoknak hívjuk. Ezek a már jól ismert jogok a Readtől a Full Controlig. Ne feledkezzünk meg a kivételekről sem. Az „Execute", „Add/remove self" vagy a „Delete tree" Active Directory jogosultságoknak nincs értelmük Exchange objektumokon.

Azok a jogosultságok, amelyekkel az Exchange telepítése bővíti az Active Directoryt, az úgynevezett Extended jogosultságok. Felsorolhatatlanul sok van belőlük, ráadásul ezek a jogosultságok objektumonként eltérnek. Például a „Create Top Level Public Folder" jogosultság a Public Folder objektumra vonatkozik, míg az „Open mail send Queue" az Exchange Server objektumán értelmezett.

Több eszközzel is megnézhetjük, illetve állíthatjuk ezeket a jogosultságokat. Megeshet, hogy ugyanazon objektum esetén is eltérő listát fogunk kapni az eltérő programokban. Egy példa erre mondjuk a First Routing Group objektuma. Más listát látunk, ha a System Managerből nézzük, és más joglistát kapunk, ha ADSIEdittel nézzük. *(Már csak zárójelben jegyzem meg, mert annyiszor volt szó róla, az ADSIEdit a Windows 2000 Support Tools része, amelynek telepítése szinte kötelező!)* Még jó, hogy ilyen sok helyről lehet állítani a jogokat. Egyébként a System Managerben az Organization, az Administrative Groups vagy a Routing Groups szintjén csak akkor látszik a tulajdonságlapok közt a „Security", ha a registryben módosítunk:

```
HKEY_CURRENT_USER\Software\Microsoft\Exchange\
EXAdmin
Value:ShowSecurityPage
Type: REG_DWORD
Data: 1
```

A „Security" lap megjelenítése a System Managerben

Ezek a jogosultságok csakúgy, mint az összes többi Active Directory jogosultság a szülőtől az alatta levő objektumokra öröklődik. Ezt mindig vegyük figyelembe, mert ez egyrészt megkönnyíti a jogok kiosztását, másrészt veszélyes, hiszen kiosszthatunk így olyan jogokat is, amiket nem kellene. Próbáljuk a jogokat úgy osztani, hogy először magasabb szinten meghatározzuk a szükséges jogosultságokat, majd az alatta levő konténereken hangoljuk finomabbra a beállításokat. A manuális jogosultságosztásnál erre még visszatérünk, konkrét példát is nézünk.

Fontos megjegyezni, ha egy objektumról levesszük a jogok öröklődését, ugyanakkor a beállított jogosultságokat nem terjesztjük le az alatta levőkre, a System Managerben többé nem tudjuk majd elérni az adott objektum alatt levő dolgokat, mert ilyenkor a „semmit nem szabad, amit nem engedünk" elv érvényesül. Tehát ha nem terjesztjük le a jogokat a szülőről a gyermekobjektumok felé, a gyermekobjektumokhoz nem lesz jogunk! Ez az implicit Deny jog. *(ADSIEdit segítségével persze még ebben az esetben is rendezhetjük a jogosultsági problémákat.)*

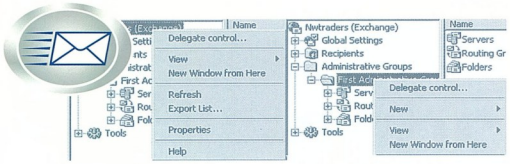
Exchange jogokra is ugyanúgy léteznek a tiltó *(Deny)* vagy engedélyező *(Allow)* jogtípusok, mint bármilyen más Active Directory, vagy NTFS jogosultság esetén. Jó példa erre a „Send As" és a „Receive As" speciális exchange jogosultság. „Send As" joggal más nevében tudunk levelet küldeni úgy, hogy a címzett nem látja a különbséget. A „Receive As" joggal mások postaládáját tudjuk megnyitni. Mind a két jog alapértelmezésben még az „Exchange Full Administrator" szereppel rendelkező felhasználóknak, ezen kívül a Domain Admins és az Enterprise Admins csoportoknak is tiltva van.

A delegáló varázsló

A rendszergazdai jogosultságok állításának egyszerű módja az Exchange Administration Delegation Wizard használata. Segítségével különböző szintű rendszergazdákat tudunk kinevezni, vagy kinevezést elvenni. Természetesen a szerepekbe nemcsak felhasználókat, hanem csoportokat is beoszthatunk.

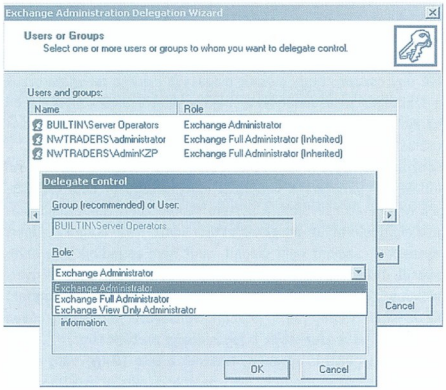
A delegáló varázsló használatához már eleve „Full Administratornak" kell lennünk, egyébként nem működik. Amelyik felhasználó nevében telepítettük az Exchange Servert, az *(és csak az!)* válik rögtön a telepítés után „Exchange Full Administrator"-rá. Persze, ha a telepítésnél csoportot adtunk meg erre a szerepre, annak a csoportnak a tagjai mind „Full Administrator" szereppel bírnak az Exchangeben.

A delegáló varázslót két helyről is lehet futtatni. Az egyik a szervezeti *(organization)*, a másik Administrative Groups szint.



A rendszergazdai jogok osztásának két szintje

Bárhonnan is indítjuk a varázslót, háromféle szerepet tudunk kiosztani. Ezek az „Exchange Full Administrator", az „Exchange Administrator" és az „Exchange View Only Administrator". Ha röviden szeretném a három szerepet megkülönböztetni, azt mondhatom, hogy a „Full Administrator" a legtöbb joggal rendelkező szerep, az „Administrator" is majdnem mindent tud, kivéve a jogosultságok állítását. A „View Only Administrator" pedig az, aki csak nézelődni tud a beállítások közt, átállítani azonban semmit sem.
Ennyire azért nem egyszerű a helyzet, mert a futtatás szintjétől függően eltérő eredményt kapunk a három szerepre.



Szerep megváltoztatása a varázslóval

A varázslóval nemcsak kiosztani tudunk szerepeket, hanem megváltoztatni az addigiakat, illetve a már kiosztott jogokat el is tudjuk venni. Nézzük meg részletesen melyik szerep milyen jogosultságokat takar!

„Exchange Full Administrator" szerep

Ha a varázslót a legmagasabb szinten futtatjuk, létrehozhatjuk az Exchange *(majdnem)* teljhatalmú urát.
Az „Exchange Full Administrator" szinte minden feladatot el tud végezni, amire szükség lehet az Exchange üzemeltetésekor. A szerephez tartozó jogok a következők:

- Full Control a „Microsoft Exchange" tárolón, és annak összes gyermekobjektumán. Ezt a tárolót nem láthatjuk a System Managerből, csak az ADSIEdittel érhető el, az alábbi útvonalon:
  CN=Configuration,DC=...
  CN=Services,
  CN=Microsoft Exchange
- Deny Receive As és Deny Send As a szervezeti *(Organization)* tárolón, és alatta az összes többi konténerben is. Ez a tiltás megakadályozza, hogy a rendszergazda hozzáférjen mások postaládájához, illetve, hogy mások nevében levelet küldjön.
- Full Control az Administrative Groups szintjén. Ez a jog fentről öröklődött, ahogy a Deny Send As és Deny Receive As jogok is.

Ha csak egy adott Administrative Group-hoz szeretnénk „Exchange Full Administrator" szerepet rendelni, onnan kell futtani a varázslót. Ilyenkor egy olyan rendszergazdát tudunk létrehozni, akinek a saját háza táján megvan minden szükséges joga, egyébként pedig láthatja a többi beállítást.
Az Administrative Groups szintjén kinevezett „Full Administrator" jogai a következők:

- Read, List object, List contents az ADSIEdittel elérhető „Microsoft Exchange" tárolón.
- Read, List object, List contents a szervezeti *(Organization)* tárolón, és alatta az összes többi konténerben is.
- Full Control, Deny Send As és Deny Receive As jogok az adott Administrative Group szintjén. Tehát ő sem tud más nevében levelet küldeni, sem más postaládáját megnyitni.
- Full Control kivéve a Change Permissions jog a „Microsoft Exchange" alatt levő „Active Directory Connection" tárolón.
- Read, List object, List contents, Write properties joga van az „Offline Address Lists" tárolóhoz.

Az „Exchange Administrator" szerep

Az „Exchange Administrator" szerep már kevesebb jogosultságot takar, a legfőbb különbség az előzőtől az, hogy az „Exchange Administrator" nem tudja állítani az objektumok jogosultságait. Ettől függetlenül szinte minden feladatot el tud végezni, amire egy Exchange üzemeltetéskor szükség lehet.
A szervezeti szinten létrehozott Administrator szereppel bíró felhasználó jogai a következők:

- Az ADSIEdittel látható „Microsoft Exchange" tárolón és minden alatta levő objektumon Full Control joga van, a Change Permissions kivételével.
- Deny Receive As és Deny Send As a szervezeti *(Organization)* tárolón, és alatta az összes többi konténerben is.
- Full Control a Change Permission kivételével az adott Administrative Group szintjén. Ez a jog fentről öröklődött, ahogy a Deny Send As és Deny Receive As jogok is.

Az Administrative Group szintjén létrehozott „Exchange Administrator" jogai:

- Read, List object, List contents az ADSIEdittel elérhető „Microsoft Exchange" tárolón.
- Read, List object, List contents a szervezeti *(Organization)* tárolón, és alatta az összes többi konténerben is.
- Minden jog a Change Permissions, a Deny Send as és Deny Receive As jogok kivételével az adott Administrative Group szintjén. Tehát ő sem tud más nevében levelet küldeni, sem más postaládáját megnyitni, sőt a jogosultságokat sem tudja átállítani.
- Full Control, kivéve a Change Permissions joga van a „Microsoft Exchange" alatt levő „Active Directory Connection" tárolón.
- Read, List object, List contents, Write properties joga van az „Offline Address Lists" tárolóhoz.

Az „Exchange View Only Administrator" szerep

A „View Only Administrator" az Exchange beállításait csak olvasni tudja, megváltoztatni már nem.
Ha a szervezeti szinten osztjuk ki ezt a szerepet, azzal mindenhová betekintést adunk a beállításokhoz. Ebben az esetben az „Exchange View Only Administrator" jogai a következők:

- Read, List object, List contents az ADSIEdittel elérhető „Microsoft Exchange" tárolón és az alatta levő összes objektumon.
- Read, List object, List contents a szervezeti *(Organization)* tárolón, és alatta a többi konténerben is.

Ha csak egy Administrative Grouphoz engedjük meg a betekintést, a következő jogokkal ruházzuk fel a felhasználót:

- Read, List object, List contents az ADSIEdittel elérhető „Microsoft Exchange" tárolón.
- Read, List object, List contents a szervezeti *(Organization)* tárolón.
- Read, List object, List contents az Administrative Groups tárolón.
- Read, List object, List contents, View Information Store az adott Administrative Group szintjén, illetve az alatta levő objektumokra
- Read, List object, List contents jog a Recipient Policies, az Address Lists, az Addressing, a Global Settings és a System Policies tárolón.

A varázsló használata valóban egyszerű, de van vele két probléma is. Az első, hogy nem lehet vele elég finoman hangolni a jogosultságokat, ezért szükség van a jogosultságok „kézi" állítására is. A második, hogy nem elég a varázsló használata, mert egyéb jogosultságok is szükségesek a tényleges feladatok elvégzéséhez. Lássuk, mit kell tenni a jogosultság varázsláson túl, ha valódi feladatokat szeretnénk megoldani!

Konkrét feladatokhoz szükséges jogok

Egy sima *(Domain Users csoportban levő)* felhasználónak is kiosztathatjuk az „Exchange Full Administrator" szerepet, mégsem lesz képes minden feladatot elvégezni, hisz egy ilyen felhasználónak nincs jogosultsága a kiszolgálót bütykölni, sőt, legtöbbször be sem tud lépni szervereken. Ugyan az Exchange System Managert nem szükséges a kiszolgálón futtatni, de tipikus feladat például a szolgáltatások újraindítása, vagy a felhasználók adminisztrációja, amelyekre egy „gyalogjúzernek" alapértelmezésben nincs joga. A varázslason túl szükséges tehát, hogy az Active Directoryban is megfelelő jogokkal rendelkezzenek az Exchange rendszergazdák, illetve a kiszolgálókon az egyéb feladatok elvégzéséhez is legyenek jogosultságaik.

Mindez a legtöbb konkrét esetben azt kívánja, hogy az Exchange rendszergazda a levelezőkiszolgálón egyben a helyi Administrators csoport tagja is legyen.

Az Exchange telepítésekor a Domain Admins és az Enterprise Admins csoportok minden jogot - a Send As és Receive As kivételével – megkapnak az Exchange matatásához. Ha olyan rendszergazdát szeretnénk kinevezni, aki az Active Directoryban és az Exchange-ben is „mindenható", legegyszerűbb, ha belerakjuk az Enterprise Admins globális csoportba. Ha az is szükséges, hogy bele tudjanak nyúlni mások postaládájába, és levelet küldhessenek bárki nevében, még meg kell szüntetni - a szervezeti szinten kiosztott - Deny Send As és Deny Receive As jogokat is. Mint ismeretes, az Enterprise Admins csoport is automatikusan tagja a helyi Administrators csoportnak, így a csoport tagjai magán a kiszolgálón is minden szükséges jogosultsággal rendelkeznek. Bizony ilyen könnyű előállítani a legtöbb joggal felruházott rendszergazdát! Nem is ez szokta okozni a problémát, hanem inkább az, amikor egy feladathoz, a legszükségesebb jogokat próbáljuk meg kiosztani. Nézzünk meg két konkrét esetet.

Postaláda-rendszergazda

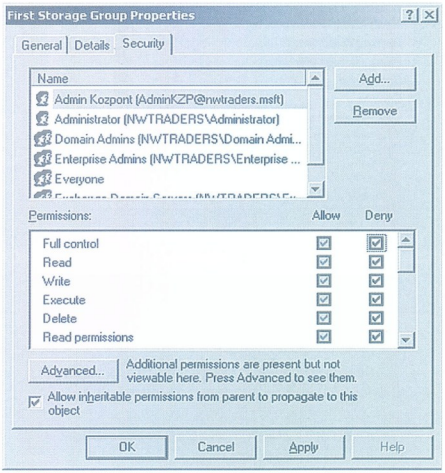
Speciális feladat például a postaládák és felhasználók létrehozása vagy törlése. A felhasználók létrehozásához minden szükséges joggal rendelkező beépített csoport az Active Directoryban az Account Operators. Ez azonban önmagában nem elég postaládák létrehozásához. Ha az Account Operators csoportnak kiosztjuk a „View Only Administrators" szerepet, máris előállt egy olyan csoport a tartományban melynek tagjai a felhasználók és postafiókok felügyeletéhez megfelelő jogokkal rendelkeznek. Nem szükséges persze a beépített csoportot használni, lehet az Active Directoryban akár szervezeti egységenként is delegálni a megfelelő jogokat, kis rendszergazdákat kinevezni. A kulcsszó ebben az esetben az Exchange oldaláról a „View Only" szerep, ami mindenképp szükséges.

Message Queue felügyelete

Az üzenetek különböző csatornákon áramlanak a kiszolgálón belül és kívül. Szükség lehet olyan rendszergazdai jogosultságok kiosztására, amelyekkel csak az üzenetek áramlását tudjuk szabályozni. Ehhez „Exchange Administrator" szerep szükséges az adott Administrative Group szintjén, valamint az adott levelezőkiszolgálón a helyi Administrators csoporttagság is elengedhetetlen. Ha csak nézelődni szeretnénk a levelezési sorokban, elég a „View Only Administrator" szerep is.

Jogosultságok „kézi" állítása

A tapasztalatok azt mutatják, a delegáló varázslóval nem tudjuk elég finoman szabályozni a jogosultságokat, így a varázsló mellett sokszor szükséges „kézzel" állítgatni szinte mindent. Ilyen helyzet például, ha nem szeretnénk, hogy egy bizonyos adatbázishoz hozzáférjen minden rendszergazda. Ennek beállításához nincs varázsló, el kell menni a Storage Group objektumhoz és a Security fülön Deny típusú jogokkal lehet megtiltani a hozzáférést.

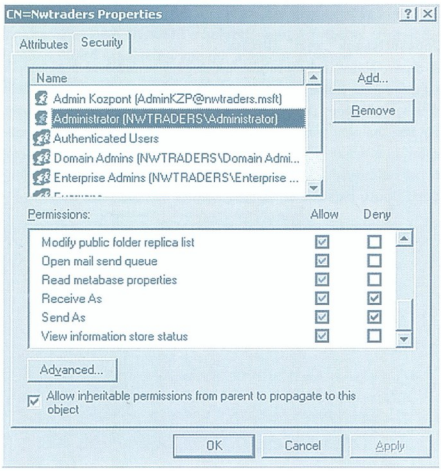


Jogok letiltása: a Deny győz a küzdelemben

De hozhatom példának a már sokszor emlegetett Send As és Receive As jogokat is. Ez a probléma úgy szokott felmerülni, hogy az egyébként minden jogosultsággal bíró rendszergazdák nem tudják megnyitni mások postaládáját, amire bizony többször szükség lehet, mint gondolnánk *(bár ezt április elseje óta öt év börtönnel díjazza a nagyeszű jogalkotó)*. Ha a varázsló segítségével osztjuk ki az „Exchange Full Administrator" szerepet, minden esetben számítsunk arra, hogy ezt a két speciális jogot még ezután „kézzel" kell megadni.

Microsoft Knowledge Base Q262054-es cikke négyféle módszert is ismertet, hogyan szabaduljunk meg a korlátozásoktól.

1. Ha a felhasználó NEM tagja a Domain Admins vagy Enterprise Admins csoportnak, hozzáadhatjuk az Exchange Domain Servers csoporthoz, és már meg is oldódott a probléma. Vigyázat! Ha egy egyszerű postaládával rendelkező – egyébként csak Domain Users csoport tagságú - felhasználót rakunk ebbe a csoportba, az is képes lesz megnyitni bárki postaládáját és levelet is tud küldeni a nevükben! Ha lehet, ezt a megoldást kerüljük!
2. Az egész Exchange szervezetre vonatkozóan úgy tudjuk biztosítani a jogot a postaládákhoz, ha az ADSIEdittel az Exchange Organization tárolón levesszük az explicite tiltást a Send As és Receive As jogokról. Ne feledjük, hogy nem elég mondjuk a – képen is látható - Administrator felhasználó esetében ezt megtenni, hanem az Enterprise Admins és Domain Admins csoportok jogait is változtatnunk kell, ha a kérdéses felhasználó az említett két csoport tagja is egyben.



A „Full Administrator" jogai az Nwtraders szervezeti szintjén

3. Egyetlen adatbázison is megszüntethetjük a korlátozásokat a System Manager segítségével. Elkattogunk az adatbázishoz – ott előhúzzuk a Security tulajdonságlapot, és megadjuk a megfelelő jogokat. Ha így járunk el, csak az adott adatbázisban levő postaládákat tudja megnyitni a rendszergazda, mert a Deny jogosultságok öröklőnek, itt csupán kivételt teszünk. Nem kell a jogosultságok öröklődését megszüntetni, elég csak a szürke Deny pipák mellé Allow pipákat tenni és már kész is vagyunk. Mindenütt azt halljuk, hogy a Deny győz az Allow jogok felett. Igen ám, de itt mégsem, mert a Deny jogokat örököltük fentről, az Allow típust pedig most pipáltuk be a Send As és Receicve As jogok mellett. Az örökölt Deny gyengébb, mint az adatbázisra konkrétan kiadott Allow, így lesz jog a postaládákhoz!
4. Directory Sites And Services MMC snap-inből is lehet állítani a postaládákhoz a jogokat. Ehhez a View menüben be kell kapcsolni a „Show Services Node" lehetőséget. Ezután akár szervezeti ***(vagy Administrative Group)***, de akár adatbázis szintjén is állíthatjuk a Send As Receive As jogokat.

A fenti négy módszer arra volt jó, hogy globálisan adjunk jogokat a postaládákhoz. Amennyiben nekünk csak egy-egy felhasználó postaládájához kell hozzáférést biztosítani, az Active Directory Users and Computersben találjuk a megoldást. Az adott felhasználó objektumán az Exchange Advanced tulajdonságlapon található Mailbox Rights gomb mögött található Full Mailbox Access jogot kiosztva biztosíthatunk jogot magunknak egy-egy postaládához.

Folyt köv.

Dorner Csilla
dorner.csilla@netacademia.net
...már egy éve MCSE 2000

.NET Akadémia

Tömbök és kollekciók rendezése, keresés, hashelés

Az előző részben áttekintettük a kollekciók legfontosabb jellemzőit, megbeszéltük a hatékonysági kérdéseket, és láttuk, hogy a tömbök sokkal okosabbak mint azt elsőre feltételeznénk róluk. Láttuk az ArrayList osztály használatát, és elkezdtünk egy formos keresési példát. Ebben a részben megbeszéljük annak részleteit, és áttekintjük a maradék kollekcióosztályok működését. A nagyobb vizuális élmény kedvéért közben összerakunk egy teljesítményszámlálókat megjelenítő alkalmazást is.

ArrayList *(folytatás)*

Az előző részt az alábbi kódrészlettel zártuk:

```
1 ArrayList names =
2 ArrayList.Adapter(lstNames.Items);
3 int pos = names.BinarySearch(txtName.Text);
4 pos = pos < 0 ? ~pos : pos;
5 pos = pos >= names.Count ? names.Count-1 : pos;
6 lstNames.SelectedIndex = pos;
```

Ez az előző számban látott alkalmazás mögötti kód, amiben egy Textboxba *(txtName)* gépelve egy ListBoxban *(lstNames)* jelöljük ki a begépelt szöveg kezdetére leginkább illeszkedő sort.
A BinarySearch pozitív számmal tér vissza, ha megtalálta a keresett elemet. Esetünkben ez a legkevésbé valószínű, hisz csak a szavak kezdetét gépelik be a felhasználók, így valószínűleg csak az összehasonlítandó stringek eleje egyezik meg. Ebben az esetben nem lenne logikus a BinarySearchnek azt mondania, hogy megtalálta a keresett elemet, így egy negatív számot ad vissza. Ha eme szám minden bitjét megfordítjuk az ellenkezőjére *(erre szolgál a ~, tilde operátor, egyes komplemens képzés)*, megkapjuk azt a pozíciót, ahová sorrendben beillik a keresendő string. Például legyen a listában:

```
batka, bodri, bubu, buksi, cicu, cirmos, jancsi,
juliska, laci, maci, micu
```

A felhasználó begépeli a „c" kezdőbetűt. Ekkor a BinarySearch -5-öt ad vissza. Ennek egyes komplemense: 4. Ez azt jelenti, hogy a „c" mint string sorrendben a 4. index elem elé illeszkedik, ami a cicu. Ezért a 6. kódsor miatt a listában a cicu lesz kijelölve. A „ci" szókezdetre a helyzet változatlan, de például „cif" szótőre már a „cirmos" a találat, mert a „cif" a „cicu" után következik.
Az 5. kódsor azt a helyzetet kezeli le, amikor egy olyan szótöredéket gépelnek be, ami sorrenden az összes listaelem mögött áll. Például „u". Ekkor túlcímeznénk a tömböt, ezért ebben az esetben ráállunk a legutolsó elemre.
Azoknak, akiknek a ?: operátor furcsa lenne, átfordítottam az előbbi kifejezéseket közönséges elágazásokra:

```
if (pos < 0)
  pos = ~pos;
if (pos >= names.Count)
  pos = names.Count - 1;
```

Eredetileg a ?: operátort a nem túl okos C fordítók miatt vezették be, a mai fordítókban egyformán gyors a kétféle konstrukció. Az előbbi tömörebb, de nehezebben olvasható, az utóbbi szószátyárabb, de a legtöbb ember számára valamivel könnyebben értelmezhető. Kinek a pap, kinek a paplan...

Hashtable

Szeretjük az ArrayListet, azonban gyakran szükségünk van valamilyen kulccsal indexelhető kollekciókra is. A .NET FCL Dictionary osztályai erre szolgálnak. Félreértés ne essék: ez nem azt jelenti, hogy a Dictionaryk csak stringkulcsokkal indexelhetők! Bármilyen típussal lehet, ami megfelelően implementálja a GetHashCode() metódust.
Legyen az a harci feladat, hogy egy százéves .ini fájl tartalmát kell feldolgoznunk, és a többiek részére a lehető legkényelmesebb módon tálalni. *(A mai világban már xml formátumú lenne egy ilyen configfájl.)* Például a C:\WINNT\Microsoft.NET\Framework\v1.0.3705\aspnet_perf.ini az ASP.NET futtató által publikált teljesítményszámláló *(performance counter)* szimbólumokat sorolja fel. Valahogy így:

```
[info]
drivername=ASP.NET_1.0.3705.0
symbolfile=aspnet_perf.h
;; ASP.NET System Counters
[text]
ASPNET_REQUESTS_QUEUED_009_NAME=Requests Queued
ASPNET_REQUESTS_QUEUED_009_HELP=The number of
requests waiting to be processed.
ASPNET_REQUESTS_TOTAL_RATE_009_NAME=Requests/Sec
ASPNET_REQUESTS_TOTAL_RATE_009_HELP=The number of
requests executed per second.
```

Azaz [] között találhatjuk a szekcióneveket, név=érték párokban pedig a szekció adatait. Minden egyes teljesítményszámlálónak van egy neve és egy magyarázó szövege. Esetünkben ezek szisztematikusan egymás után jönnek, bár ezt a feldolgozáskor nem fogjuk kihasználni. A ;-vel kezdődő sorok kommentek, azokat általában nem érdemes feldolgozni.
A feldolgozáskor csak egy szekcióra vagyunk kíváncsiak, ezért az egyszerűség kedvéért csak azzal foglalkozunk, a többi adatát átlépkedjük. A megvalósítandó feldolgozóosztály interfésze így fog kinézni:

```
public interface IIniReader {
  void LoadIniFile(string filePath,
```

```
                                    string sectionName);
        Hashtable GetSectionEntries();
    }
```

Azért definiáltam egy interfészt a feldolgozáshoz, mert létezhetnek különböző formátumú .ini fájlok, amelyekhez természetesen különböző értelmezőosztályokat kell írni. A felhasználói programunk az interfészen keresztül fog valamely konkrét implementációval dolgozni, de nem kell tudnia arról hogy az milyen típusú .ini fájlt olvas fel. Így a későbbiekben az .ini szerkezetének megváltozása nem lesz látható, nem kell átírni a felhasználó programokat, csak más implementáló osztályból kell egy példányt létrehozni. Ez az interfészalapú programozás egyik legnagyobb előnye. OOP terminológiával élve csökkenti a csatolást a hívó és a hívott osztályok között.

A LoadIniFile() működése a következő:

- soronként felolvassuk az .ini fájlt
- levágjuk a sor kezdő és záró whitespace karaktereit
- átlépjük az üres sorokat
- a ;-vel kezdődő sorokat eldobjuk
- addig olvasunk üresen, amíg el nem érjük a fájl végét, vagy nem találunk egy [szekciónév] kezdetű sort *( ] után még lehet komment)*
- megjegyezzük, hogy a kívánt szekcióban vagyunk, és addig olvassuk a sorokat, amíg egy újabb szekció nem kezdődik *([a sor elején)*
- a kapott sorokat kettévágjuk a ; mentén, csak az eleje érdekes *(a végén komment is lehet)*
- a kettévágott sor első felét kettévágjuk az = mentén
- a darabok végeiről levágjuk a whitespace-eket
- a darab első felét mint kulcsot használva elmentjük a Dictionarynkbe a második felet, a kulcshoz tartozó értéket

Lássuk hogyan néz ez ki C#-ra lefordítva. A tömörség kedvéért a megjegyzéseket eltávolítottam a kódból, de az **[1]** címen megtalálható a teljes, rendesen kommentezett forrás.

```
public class WindowsIniReader: IIniReader {
  private enum IniReaderState {
    InSection,
    OutSection
  }
  private string filePath;
  private string sectionName;
  private bool foundSection = false;
  private Hashtable sectionData;
  private StreamReader reader;

  public WindowsIniReader() {
    sectionData = new Hashtable();
  }

  #region Implementation of IIniReader
  public bool LoadIniFile(string filePath,
                          string sectionName) {
    this.filePath = filePath;
    this.sectionName = sectionName;
    try {
      reader = new StreamReader(filePath);
      LoadIniFileImpl();
      return foundSection;
    }
    finally {
      if (reader != null)
        reader.Close();
    }
  }

  public Hashtable GetSectionEntries() {
    if (foundSection)
      return sectionData;
    else
      return null;
  }
  #endregion

  private void LoadIniFileImpl() {
    string line;
    string sectionString = "[" +
            sectionName + "]";
    IniReaderState state =
      IniReaderState.OutSection;

    while ((line = reader.ReadLine()) != null) {
      line = line.Trim();
      if (line == "" || line[0] == ';')
        continue;

      switch(state) {
        case IniReaderState.OutSection:
          if (line.StartsWith(sectionString)) {
            state = IniReaderState.InSection;
            foundSection = true;
          }
        break;
        case IniReaderState.InSection:
          if (line.StartsWith("[")) {
            state = IniReaderState.OutSection;
          }
          else {
            line =
              line.Split(new char[] {';'}, 1)[0];
            string[] namevalue =
              line.Split(new char[] {'='}, 2);
            if (namevalue.Length == 2) {
              sectionData.Add(
                namevalue[0].Trim(),
                namevalue[1].Trim());
            }
          }
        break;
      }
    }
  }
}
```

Olvastassunk fel egy .ini fájlt az előbbi osztály használatával! A fájlnevet bekérjük a felhasználótól (WinForms alkalmazás):

```
Hastable perfNames;
OpenFileDialog fd = new OpenFileDialog();
fd.Filter =
  "Ini fájlok(*.ini)|*.ini|Minden fálj(*.*)|*.*";
if (fd.ShowDialog() == DialogResult.OK) {
  IIniReader reader = new WindowsIniReader();
```

```
    reader.LoadIniFile(fd.FileName, "text");
    perfNames = reader.GetSectionEntries();
}
```

Látható, hogy a WindowsIniReader osztályból hozunk létre egy példányt, de azt az IIniReader interfészen keresztül ragadjuk meg. Így más .ini olvasó implementációk használatához csak a new után kell kicserélni az osztály nevét, semmi más teendőnk nincs.
Hogyan érjük el a beolvasott számlálók jellemzőit? Például szaladjunk végig az összes beolvasott elemen, és szedjük ki a számunkra fontosakat:

```
foreach(DictionaryEntry perf in perfNames) {
   string symbol = (string)perf.Key;
   string text = (string)perf.Value;
   //Az ASPNET_ kezdetűek a tényleges
   //teljesítményszámlálók, a 000 jelzi a
   //nyelvsemleges verziót
   if (symbol.StartsWith("ASPNET_") &&
       symbol.EndsWith("_000_NAME")) {
}
```

Mi az a DictionaryEntry? Egy Dictionary-t bejárva mindig név-érték párosokat kapunk vissza. Ilyen párost egy egyszerű típus nem tud eltárolni, ezért pont a bejáráshoz definiálták ezt az osztályt. Látható, hogy a Key jellemzőn keresztül kiolvasható az aktuális elem kulcsa, míg a Value-n keresztül az értéke.
Ha ismerjük valamely kulcs értékét, akkor az alapján könnyedén meghatározható a hozzá tartozó tárolt érték:

```
MessageBox.Show((string)perfNames[
   "ASPNET_REQUEST_EXECUTION_TIME_000_NAME"]);
```

Olyan, mint a sima tömb, csak nem egészekkel, hanem most egy stringgel címeztük meg a tartalmát. Mind a kulcs, mind a tárolt érték általános object típusú. Emiatt kiolvasáskor állandóan stringgé kellett konvertálnunk a visszakapott értéket. Ez felesleges teljesítményveszteséget okoz, és még kényelmetlen is. Ha kifejezetten stringeket kell tárolni stringkulcsokkal címezve *(nálunk is ez a helyzet)*, jobban használható a StringDictionary osztály. Ebben a kulcsok és az értékek is stringek, de vigyázzunk arra, hogy a kulcsok összehasonlítása case insensitive *(kis-nagybetű érzéketlen)* módon történik.
Az előbbi példánkat befejezendő jó lenne megjeleníteni a felolvasott teljesítményszámlálók értékét. Minden egyes teljesítményszámláló valamilyen kategóriába tartozik. Ezeket hívja a Performance Monitor Performance Objectnek. Az .ini fájl [objects] szekciója felsorolja ezeket, ám a számlálók és a kategóriák összerendelését csak a kommentezett sorok vizsgálatával tehetjük meg, de ezt sem lehet megtenni tisztán gépi úton. Emiatt, listánk alapján csak csúnya hekkeléssel lehet megjeleníteni a számlálókat. Természetesen erre a célra a Performance* kezdetű osztályok vannak kitalálva, azokkal végig lehet gyalogolni és kiolvasni a számlálókat. A megjelenítő WinForms alkalmazás letölthető a **[1]** címen.
A példában megfigyelhető hogyan kell nyújtható ablakot létrehozni, hogyan kell egy szál prioritását állítani, vezérlőket dinamikusan létrehozni, kurzort módosítani, na és felhasználjuk az előbbi .ini olvasó osztályt is.

SortedList

A SortedList a Hashtable és az ArratList hibridje. Kulcs-érték párokat tárol mint a Hashtable, de az elemek elérhetők egész indexekkel is. A lista belülről és kívülről nézve is rendezett, de nem a kulcsok hashkódja *(ami kívülről teljesen haszontalan)*, hanem maguk a kulcsok alapján. Ehhez természetesen a kulcsoknak implementálni kell az IComparable interfészt, ahogyan az előző részben láttuk.
Nézzünk a használatára egy példát. Nosztalgiából felolvastatjuk a system.ini 386enh szekcióját, először egy Hastable-be:

```
IIniReader reader = new WindowsIniReader();
reader.LoadIniFile(@"C:\WINNT\system.ini",
                   "386enh");
Hashtable ht = reader.GetSectionEntries();
ShowDictionary(ht);

static void ShowDictionary(IDictionary dick) {
  foreach(DictionaryEntry ent in dick) {
    Console.Write("{0}: ", ent.Key);
    Console.WriteLine(ent.Value);
  }
}

EGA40W0A.FON: EGA40W0A.FON
FileSysChange: off
CGA80W0A.FON: CGA80W0A.FON
EGA40W0A.FON: EGA40W0A.FON
EGA80W0A.FON: EGA80W0A.FON
woafont: dosapp.FON
```

A kimenet teljesen véletlennek ható sorrendben látható, valójában az első oszlop *(kulcs)* hashkódjai szerint van rendezve, amelyet a string.GetHashCode() generál a háttérben.
Figyeljük meg, hogy a kollekció bejárásához definiált metódusban *(ShowDictionary())* nem Hastable referenciát várunk, hanem IDictionary-t. Mivel a SortedList és a Hashtable is implementálja ezt az interfészt, egységesen tudjuk őket kezelni.
Töltsük át a Hashtable tartalmát a SortedListbe, és listázzuk ki a tartalmát:

```
SortedList sl = new SortedList(ht);
ShowDictionary(sl);

CGA40W0A.FON: CGA40W0A.FON
CGA80W0A.FON: CGA80W0A.FON
EGA40W0A.FON: EGA40W0A.FON
EGA80W0A.FON: EGA80W0A.FON
FileSysChange: off
woafont: dosapp.FON
```

Látható, hogy a két lista közötti átjárás nem túl bonyolult *(megint csak az IDictionary miatt)*. A kimenet ezúttal a kulcsok szerint rendezve jött elő. Nézzünk olyan tartalmat, amiben a kulcsok kis-nagybetűben különböznek:

```
sl.Clear();
sl.Add("maci", "a medve");
sl.Add("laci", "szintén medve");
sl.Add("Maci", "a medve");
sl.Add("Laci", "szintén medve");
ShowDictionary(sl);

laci: szintén medve
Laci: szintén medve
maci: a medve
Maci: a medve
```

Látható, hogy stringkulcsok esetén az összehasonlítás, így a rendezettség is nem érzékeny a kis-nagybetűre, így került az „l" után az „L", illetve az „m" után a „M". A kulcsok közötti egyenlőség vizsgálata kis-nagybetű érzékeny, máskülönben a SortedList nem engedte volna meg a „laci" után berakni a „Laci"-t, hisz a kettőt egyenlőnek tekintené, és egy Dictionary-ban nem lehetnek duplázott kulcsok.
Ha nekünk nem megfelelő ez a működés, a konstruktorban átadhatunk egy IComparer implementációt, ami kis-nagybetű érzéketlen módon hasonlít össze két stringet. Ilyen kész megvalósítás a CaseInsensitiveComparer osztály:

```
SortedList sl = new SortedList(
  new CaseInsensitiveComparer());
```

Alapértelmezésben a SortedList a Comparer osztály Default jellemzőjén keresztül kér le egy kis-nagybetű érzéketlen IComparer megvalósítást, azaz koncepcionálisan ez történik benne:

```
SortedList sl = new SortedList(Comparer.Default);
```

Próbáljunk meg berakni csak kis-nagybetűben különböző kulcsokkal elemeket a CaseInsensitiveComparer használata mellett:

```
sl.Add("maci", "a medve");
sl.Add("Maci", "a medve");
System.ArgumentException: Item has already been
added. Key in dictionary: "maci" Key being added:
"Maci"
```

A SortedList belülről két tömböt tartalmaz: egyet a kulcsoknak, amelyet a módosítások során is rendezve tart, egy másikat az értékeknek. Így a kulcs alapján ebben is gyorsan lehet keresni, habár nem annyira gyorsan, mint a Hastable-ben, hisz itt a bináris keresésnek nem 32 bit hosszú hashkódokat kell összehasonlítani, hanem az eredeti méretű kulcsokat:

```
//Keresés kulcs alapján (közepesen gyors)
Console.WriteLine(sl["laci"]);
a medve
```

A kettős belső kialakítás célja, hogy az elemeket el lehet érni egész indexekkel is. Ehhez azonban nem a szögletes zárójeles *(indexer)* formátumot használjuk, mert azon keresztül a kulcs alapján lehet kiolvasni az értékeket, hanem a GetKey() és a GetByIndex() metódusokat:

```
//Keresés index alapján (nagyon gyors)
Console.WriteLine(sl.GetKey(2));
maci
Console.WriteLine(sl.GetByIndex(2));
a medve
```

Az indexekkel történő elérés természetesen extrém gyors, hisz belül egyszerű tömbelérés történik.
Az értékek alapján is kerestethetünk:

```
//Keresés értékek alapján (lassú)
Console.WriteLine(
  sl.ContainsValue("szintén medve"));
True
Console.WriteLine(sl.ContainsValue("blabla"));
False
```

Az értékek szerinti keresés átmegy lineáris keresésbe, így nyilvánvalóan ez a leglassabb keresési módszer a SortedListben.

BitArray

Egyszerű, de időnként jól használható osztály a BitArray. Belülről bitenként ábrázol egy igen/nem értéket, azaz kívülről egy bit értékét boolként láthatjuk.

```
//41 bit hosszú bittömb, minden elem 0 értékű
BitArray barr = new BitArray(41, false);
//Beállítjuk a 23. bit értékét 1-be
barr.Set(22, true);
//Megfordítjuk a biteket
barr.Not();
//Kiíratjuk a biteket
foreach(bool bit in barr) {
  Console.Write(bit ? 1 : 0);
}
Console.WriteLine();

11111111111111111111101111111111111111111
```

Kicsit talán furcsa, hogy az IEnumerable interfész segítségével *(a foreach használja)* be lehet járni egy bittömböt, de miért is ne?

Típusos kollekciók

Ha típusos kollekcióra van szükségünk, viszonylag keskeny pallón kell mozognunk.
Stringekhez találunk némi segítséget. A System.Collections.Specialize névtérben a StringCollection olyan, mint az ArrayList, csak stringeket tárol típusos módon. A StringDictionary a Hashtable-höz hasonlít, csak stringek a kulcsok és a tárolt értékek is. A NameValueCollection hasonló a SortedListhez, de minden egyes string kulcshoz több string értéket is tud tárolni. Ilyenekkel gyakran találkozunk, például gondoljunk az alábbi http querystringre:

```
x.aspx?list1=piros&list1=barna&edit1=kukac
```

Ebben egy kulcshoz *(list1)* több érték is tartozik *(piros, barna)*. Az ilyen jellegű információk feldolgozására találták ki a NamedValueCollectiont.

```
NameValueCollection nvc = new
  NameValueCollection();
nvc.Add("list1", "piros");
nvc.Add("list1", "barna");
string[] elemek = nvc.GetValues("list1");
Console.WriteLine(elemek[0]);  //piros
Console.WriteLine(elemek[1]);  //barna
```

Egyéb típusos kollekciók írásához nyújtanak segítséget a CollectionBase, DictionaryBase, és a NameObjectCollectionBase osztályok. Ha érték szerinti típusokat akarunk konverziók és boxolás nélkül tárolni, a **[1]** címen a STColl.cs-ben található megoldás osztály nyújthat segítséget.

Soczó Zsolt MCSE, MCSD, MCDBA
Zsolt.Soczo@netacademia.net

A cikkben szereplő URL-ek:
[1]: Letölthető példakódok
http://technet.netacademia.net/download/dotnet

AZ Exchange 2000 Server és Web Storage System: *Event Sinkek*

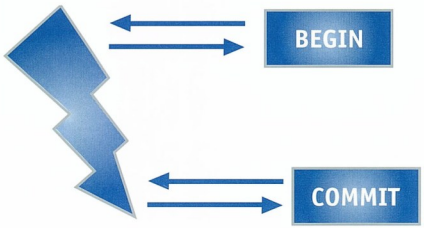
Véleményem szerint az egyik legizgalmasabb dolog az informatikában az, amikor a dolgok „maguktól történnek". Levelek jönnek-mennek, dokumentumok megváltoznak, mintha valami eleven kobold ülne a gépünk belsejében. A kobold megszelídíthető, sőt engedelmességre is kényszeríthető, csak meg kell tanulnunk, hogyan szóljunk hozzá. A jelszó: Event Sink!

Ami a háttérben van

A háttérben a Web Storage System Event Sinkjei lapulnak. Ezek olyan COM+ komponensek, amelyek beregisztrálásuk után folyamatosan figyelik a WSS-t. Amennyiben olyan esemény történik, amely őket érinti, lefutnak, végrehajtják előre definiált feladatukat: például abortálják a műveletet, értesítést küldenek adott személynek, adatokat mozgatnak a WSS-en belül stb. Négy alapvető típust különböztetünk meg:

- szinkron események
- aszinkron események
- rendszeresemények
- időzített események

A **szinkron események** végrehajtása további három fázisra bontható: BEGIN, COMMIT és ABORT. A BEGIN-re akkor kerül sor, amikor már tudjuk, mi lesz az esemény, amire reagálunk, azonban az még nem indult el. Ily módon minden elkapható, még mielőtt bármi történne. Így akár le is tilthatjuk a műveletet, anélkül, hogy az bármily apró dolgot tenne *(például dokumentum törlésekor lehet hasznos ez a funkció)*. Ha az esemény rendben lefutott, a futás az Event Sink COMMIT ágára kerül, mielőtt visszaadná a vezérlést. Ha az esemény meghiúsult, az ABORT ágat hajtja végre a rendszer.



Sikeres esemény lefolyása szinkron Event Sinkkel

Aszinkron Event Sink esetén az általunk előírt folyamat az esemény bekövetkezte után hajtódik végre, meghatározatlan idő eltelte után. Ez az idő sok mindentől, például a processzor terheltségétől, egyéb eseményektől függ. Pontosan az eltelt idő határozatlansága miatt képes látszólag determinisztikus dolgokat művelni, amelyek fejre állíthatják még a legtapasztaltabb programozót is. Erről még később szólok.

A **rendszeresemények**, mint azt a nevük is mutatja, a rendszer állapotára reagálnak: a rendszer indulásakor illetve leállításakor futnak le.

Az **időzített események** *(gondolom, nem meglepő módon)* bizonyos, általunk előre definiált időközönként futnak le. Például archiváláskor lehetnek hasznosak.

Event Sinkek létrehozása

Most lássuk, hogyan hozhatunk létre egy egyszerű Event Sinket. Tegyük fel, azt szeretnénk megvalósítani, hogy a WSS minden új e-mail érkezésekor küldjön értesítést erről a controller nevű felhasználónak. Ezt nyilván aszinkron módon érdemes megvalósítani, mert nem szükséges a beavatkozás menet közben.

```
Option Explicit
Implements Exoledb.IEXStoreAsyncEvents

Private Sub IEXStoreAsyncEvents_OnDelete (...)
End Sub

Private Sub IEXStoreAsyncEvents_OnSave (ByVal
pEventInfo As Exoledb.IexStoreEventInfo, ByVal
bstrURLItem As String, ByVal lFlags As Long)
  With new CDO.Message
    .From = "felado@encegem.hu"
    .To = "controller@encegem.hu"
    .Subject = "Uj e-mail erkezett"
    .HTMLBody = "<b>Uj e-mail erkezett.
     Helye:</b>" & bstrURLItem
    .HTMLBody = .HTMLBody + "<br><b>Datum: </b>"
    .HTMLBody = .HTMLBody + Date + " " + Time
    .Send
  End With
End Sub
```

Mi is történik itt? Az OnSave metódus akkor kerül végrehajtásra, amikor a levél már beérkezett a megfelelő mailboxba, vagyis WSS mappába. Aszinkron módon vizsgáljuk ennek bekövetkezését, tehát pontosan nem tudhatjuk, mikor kerül sor az eseménykezelő futására, csak abban lehetünk biztosak, hogy addigra a levél már beérkezett a címzetthez.

Az e-mailt CDO segítségével küldjük: beállítjuk a feladót, a címzettet, a levél tárgyát, és törzsét, majd a .Send metódus segítségével elküldjük. Ennyi az egész...

Másik jó példa lehet az, ha nem akarjuk engedni, hogy bármit is töröljenek egy számunkra fontos mappából. Ez szinkron Event Sink lesz, hiszen bele akarunk avatkozni az esemény lefolyásába, azt akarjuk, hogy az ne legyen végrehajtható.
Az ehhez tartozó programkód rendkívül egyszerű:

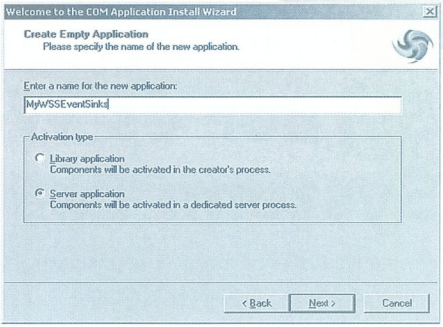
```
Option Explicit
Implements Exoledb.IEXStoreSyncEvents

Private Sub IEXStoreSyncEvents_OnSyncDelete
(ByVal pEventInfo As Exoledb.IexStoreEventInfo,
yVal bstrURLItem As String, ByVal lFlags As Long)
  If lFlags And EVT_SYNC_BEGIN Then
    Dim iEventInfo As
      Exoledb.IExStoreDispEventInfo
    Set iEventInfo = pEventInfo
   iEventInfo.AbortChange
  End If
End Sub

Private Sub IEXStoreSyncEvents_OnSyncSave (...)
End Sub
```

Szinkron eseménykezelőről lévén szó, az esemény végrehajtásába történő beavatkozást már az esemény bekövetkezése előtt meg kell tennünk. Ennek vizsgálata történik az **if** sorban, az **lFlags** segítségével. Az esmény egyes fázisai:

EVT_SYNC_BEGIN	Az esemény kezdeti állapota, amikor még semmi nem történt
EVT_SYNC_COMMIT	Az esemény sikeresen lefutott
EVT_SYNC_ABORT	Az esemény lefutása valamilyen okból meghiúsult

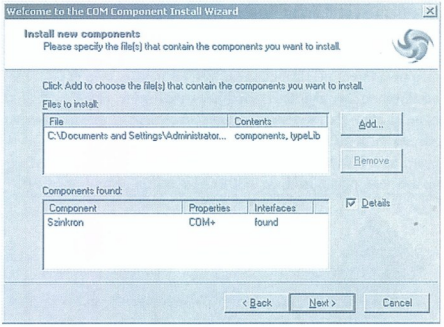
A blokk belsejében létrehozunk egy iEventInfo nevű változót, amely az **IExStoreDispEventInfo** interfészt implementálja. Ennek átadjuk a paraméterként kapott **pEventInfo** értékét, és ezen hajtjuk végre az **AbortChange** metódust.

Megírtuk tehát a kódokat, most készítsünk ezekből DLL-t. Ha ezzel is megvagyunk, jöhet a COM+ regisztráció: a Component Services-t megnyitva *(Start menü, Administrative Tools)* hozzunk létre a gépen saját applikációt: Component Services/Computers/My Computer/COM+ Applications/ jobb egérgombbal New Application. Az „Install or Create a New Application" ablakban válasszuk a „Create an empty application" lehetőséget. Adjunk neki egy számunkra megfelelő nevet, majd jelöljük ki, hogy Server application-t szeretnénk létrehozni.



Új szerveralkalmazás létrehozása

A következő ablakban azt adhatjuk meg, hogy mely felhasználó jogaival fusson majd az eseménykezelőnk. Kijelölhetjük, hogy az éppen aktuálisan belépett felhasználó nevében, vagy például konkrétan Béla jogosultságaival férjen hozzá az Event Sink a WSS-hez. Az utolsó ablakban klikkeljünk a Finish gombra, és már kész is a saját COM+ alkalmazásunk. Már csak azt kell megmondani neki, hogy tulajdonképpen mit is csináljon.
A Component Services ablakban álljunk rá a MyWSSEventSinks/Components csomópontra, majd jobb egérgombbal történő klikkelés után jelöljük ki a New Component menüpontot. Ha kiválasztjuk az Install New Component(s) lehetőséget, megadhatjuk a DLL-ünk elérhetőségét. A következő ablakban láthatjuk a felvett komponenst, és újabbakat is adhatunk hozzá.



Új komponens felvétele

Megvan a DLL, beregisztráltuk, most már csak azt kell megmondani, hogy hova csücsüljön a koboldunk, vagyis hogy mely WSS-mappákat figyelje. Ezt legegyszerűbben az Exchange Explorer segítségével tehetjük meg. Kapcsolódjunk a kívánt mappára, majd a Detail View nézetben jobb egérgombbal kattintsunk az Items-re, és válasszuk ki az Event Registration Wizard menüpontot.

A varázsló a következő dolgokra kér bennünket: először is meg kell adnunk az Event Sink nevét. A következő ablakban azt kell definiálnunk, hogy milyen típusú eseménykezelőt kívánunk éppen beregisztrálni: szinkron, aszinkron, rendszer, vagy időzített. Ennek megfelelően a következő lépés az, hogy megadjuk, mely metódusok lényegesek számunkra (*hiszen a dll-nek tartalmaznia kell az összes, adott típushoz tartozó metódust, még ha azok törzse üres is)*. Az Event Sink scope-ja lehet deep, exact vagy shallow, annak megfelelően, hogy az alatta- és fölötte lévő mappákat akarjuk-e vizsgálni. Végül meg kell adnunk, hogy beregisztrált COM komponenst, vagy scriptet szeretnénk futtatni, és meg kell adni ennek elérését is.

Jöhet a próba! Hozzunk létre egy új elemet a megfelelő mappában *(akár közvetlenül az Exchange Explorerből, akár más alkalmazásból)*, és figyeljük a controller leveleit! Néhány másodpercen belül megérkezik az értesítés: valami történt...

Most próbáljuk meg törölni ezt az elemet, először az Explorerből. A törlés lehetetlen! Lássuk, mi történik, ha a furfangos felhasználó a WSS-t fájlrendszernek tekinti, és az M: meghajtóról, Windows Explorerből kísérli meg a törlést. Az e-mailt sikerült kitörölni! Most akkor mire is jó ez az egész?!

Álljunk meg egy pillanatra, és nézzünk szét egy kicsit jobban: az Exchange Explorerhez visszatérve frissítsük a nézetet, és lám: az általunk törölt elem ott van a helyén! A Windows Explorerben nincs! Mi ez az egész?!

A WSS 20 percenként aktualizálja a tartalmát

Most menjünk el kávézni, akár ebédelni is, a lényeg, hogy legalább 20 percig hagyjuk békén a WSS-t. A pihenő közben gondoljunk valami másra, ne ezen törjük a fejünket, úgyis hamarosan választ kapunk mindenre!

Ha eltelt a megfelelő idő, frissítjük a Windows Explorert, és csodát láthatunk: a törölt elem visszakerült a helyére! Most akkor tényleg: mi is történik itt? A fájlként törölt elem a WSS-ben ott maradt, hiszen onnan nem törölhetjük ki – épp az előbb tiltottuk meg. A WSS 20 percenként aktualizálja a tartalmát, ezért láthattuk úgy, mintha az elem eltűnt volna, és ezért került vissza, amíg mi kávézgattunk. Nem szomszéd kollégánk tréfált meg bennünket!

Event Sinkek felülírása

Ha menet közben derül ki, hogy az Event Sinkünket egy kicsit még át kellene írni, de már beregisztráltuk, és szépen működik, semmi pánik, nem kell mindent elölről kezdenünk. Elég leállítani a COM+ objektumot *(Component Services, Shut down)*, újrafordítani a DLL-t, és már működik is!

Levelek attachmenttel

Most lássunk egy kis csemegét.

Nemrégiben bemutattam, hogyan lehet létrehozni e-maileket, illetve csatolmányokat fűzni hozzájuk. Most oldjuk meg azt eseménykezelők segítségével, hogy ha egy bizonyos mappába érkeznek e-mailek, automatikusan átkerüljenek egy másik mappába, például azért, mert a címzett a hr@encegem.hu, és szeretnénk, ha ez automatikusan eljutna az éppen aktuális HR vezetőhöz, mondjuk a gizike@encegem.hu címre.

Mit lehet tenni? Próbáljuk ki mindkét lehetőséget, először tehát kísérletezzünk szinkron módon. Nyilván a levelet akkor kell átmozgatni, amikor már bekerült a helyére, tehát az esemény bekövetkezése után: COMMIT.

Az Event Sink regisztrálása után Gizike szépen kapja a leveleket, megnyugodhatunk.

Igen ám, de egyszer csak egy furfangos felhasználó az önéletrajzát és fényképét is mellékeli a levélhez, és HR-es munkatársunk sikítva tör ránk: nem kapja meg a csatolt fájlokat a levelekkel!

Nem baj – gondoljuk –, biztosan azzal van baj, hogy szinkron eseménykezelőt használunk, és talán a WSS mélyén élő koboldok tréfálkoznak. Majd mi kifogunk rajtunk! Tegyük át az egészet aszinkron módba. Miután ezzel készen vagyunk, már óvatosabban járunk el, és elkezdjük próbálgatni, mi történik, ha csatolt állomány is van a levél mellett.

Borzalom: ezek most sem mozdulnak el a levéllel, útközben valahol elvesznek. Vagyis... Ha kellően sokszor próbálkozunk, azt tapasztalhatjuk, hogy időnként mégis megérkeznek. Időnként. Először nyilván arra gondolunk, hogy talán a csatolmány típusától, darabszámától stb. függ az átvitel sikeressége, de azt tapasztaljuk, hogy nem. A dolog teljesen determinisztikus, és viszonylag kis százalékban sikeres az átvitel.

Mindezekből a következő tanulságot vonhatjuk le: a kódunk feltehetően nem rossz, és vinné magával a csatolmányokat is. De valami közbeszól, valami, ami mélyebben rejtőzik...

Nem csigázom tovább a kedélyeket, elárulom a megoldást: korábban láthattuk, hogy az e-mailek létrehozása, és a csatolmányok hozzáfűzése két külön lépés.

Amikor szinkron eseménykezelőt használtunk, a COMMIT-ot vizsgáltuk. Mikor is következik ez be? Először akkor, amikor az e-mailt létrehozzuk, majd még egyszer, amikor hozzáfűzzük a kívánt állományt. Az eseménykezelő az első COMMIT észlelésekor végrehajtja a megfelelő kódot, és áthelyezi az e-mailt. A csatolmányt már nincs is hova mentenünk, hiszen az e-mail már nem is létezik ekkor, „kirántottuk" alóla!

Aszinkron esetben néha működik a kód. Lássuk, mi az oka ennek. Az aszinkron eseménykezelők akkor futnak le, amikor az esemény már befejeződött, de nem tudni, pontosan mennyi idővel utána. Ha a gépünk terheltsége kicsi, jó eséllyel nagyon rövid időn belül. Kellően hamar ahhoz, hogy a csatolmányokat még ne is hozzuk létre! Ha az eseménykezelő egy kicsit később jut szóhoz, addigra már hozzáfűztünk az e-mailhez mindent, amit szerettünk volna, így ezek is áthelyeződnek vele együtt.

Furfangos dolgok ezek. Sok időbe és programozói verejtékbe kerül, amíg pontosan kitapasztaljuk, hogyan is lehet megszelídíteni a koboldokat. De megéri, mert azután már készségesek és segítőkészek.

Molnár Ágnes
agnes.molnar@t-systems.com

Patchwork: Javítások Excelhez Wordhöz

Új javítócsomag jelent meg az Excel 2000, Excel 2002 és a Word 2002-es Microsoft programokhoz, mely az MS02-031-es számot viseli **[1]**. Ezek a javítások az eddig ismert hibákon kívül négy új biztonsági rést tömnek be.

Az egyik biztonsági rés az Excel makrókezeléséből adódik. Ha a munkafüzetbe olyan objektumokat ágyazunk, melyekhez makrók kapcsolódnak, ezek a kódok automatikusan lefutnak, kikerülve az Excel makrókra vonatkozó biztonsági beállításait.

Szintén Excel makrók futtathatók automatikusan, ha egy munkafüzetet egy hiperlinken keresztül nyitunk meg. A futtatható utasítások itt is éppúgy átcsúsznak a biztonsági beállításokon mint az előző esetben.

HTML oldalakba ágyazott scriptek futtathatók a helyi zónában *(így valószínüleg alacsonyabb védelmi beállítások mellett)*, ha a scriptet egy XSL stylesheetbe ágyaztuk és ezt Excelben megnyitjuk.

A Word körlevelek egy újabb biztonsági rését is megtalálták. Ez a változat lehetővé teszi, hogy a rosszindulatú támadók a védelmi beállításokat kikerülve automatikusan futtassa kódját. Ez azonban csak akkor lehetséges, ha a rendszerünkön megtalálható a Microsoft Access és a körlevelet HTML formátumba menettük el.

Ezeken a hibákon kívül persze még egyéb hibák javításait is tartalmazzák az update-ek, ezek közül szemezgetünk majd néhányat. Mielőtt azonban hozzáfognánk, néhány telepítéssel kapcsolatos információról lesz szó. Mindhárom javítás esetén elengedhetetlen, hogy az update telepítése előtt bizonyos javítócsomagokkal rendelkezzünk. Excel 2000 esetén ez az Office 2000 SR1, Excel 2002 és Word 2002-nél pedig a Microsoft Office XP SP1 az előfeltétel.

A javítások telepítése után azokat már nem tudjuk eltávolítani, sőt, ha újra próbálkozunk a telepítéssel, rendszerünk közli, hogy a javítócsomag már rendszerünk része. Azt, hogy a telepítés sikeresen befejeződött, az adott program verziószámán ellenőrizhetjük. Az új verziószámok a következők:

Excel 2000 esetén: 9.0.6508
Excel 2002 esetén: 10.0.4109.0
Word 2002 esetén: 10.0.4109

Végül nézzünk egy-két hibát, melyeket az adott javítással kiküszöbölhetünk.

Excel 2000 SR-1 Update

- Ha egy fájlt bináris szinten megváltoztattak, előfordulhat, hogy átcsúszik a macro védelmi beállításain.
- Amikor egy olyan Excel fájlt nyitunk meg, melyet egy víruskergető már fertőzöttként regisztrált, előfordulhat, hogy az Excel nem figyelmeztet minket, hogy a fájl makrókat tartalmaz. Így „tudtunkon kívül" engedélyezzük a további támadásokat rendszerünk ellen.
- Ha pár másodpercnél tovább tartjuk tartva a bal egérgombot, a CPU leterheltsége felugrik 100%-ra.
- Ha egy Excel fájlt egy 255 karakternél hosszabb URL-lel nyitunk meg, előfordulhat, hogy a következő hibaüzenettel találjuk magunkat szembe:
  File name is locked for editing by 'user name'.

Excel 2002 Update

- Ha az Excelt terminálon keresztül használjuk és onnan nyomtatunk, a nyomtatandó dokumentum minden aktív terminálablak nyomtatójára megjelenik.
- A digitális aláírás törlődik a dokumentumból, ha automatikus mentés történik.
- Amikor egy Excel dokumentum olyan makrót tartalmaz, amely a dokumentum megnyitásával automatikusan elindulna és ezt mi nem engedélyezzük, az Excel azonnal bezáródik és "-2147417848"-as számú hibát generál.
- Amikor egy olyan Excel Web lekérdezést *(.iqy)* szeretnénk megnyitni, amely egy hosszú hivatkozást tartalmaz, az Excel automatikusan bezáródik.
- Egy előző javításból származó hibát is javíthatunk. Ha egy olyan Excel dokumentumot nyitunk meg, amely beágyazott objektumot tartalmaz és a gépünkön antivírus program is fut, az adott objektumot nem lehet aktiválni.
- Amikor olyan adatokat törlünk a munkafüzetünkből, amelyekre egy diagram hivatkozik, majd a diagramot szerkeszteni szeretnénk, az Excel automatikusan bezáródik.

Word 2002 Update

- Ha ESC-t nyomunk a nyelvi ellenőrzés ablakában, miközben a szótárnyelvek között válogatunk, előfordulhat, hogy lefagy a Word.
- Ha a Word dokumentumunkat levélként küldjük, és előzőleg engedélyeztük a TrueType betűtípusok beágyazását, a Word drasztikusan megnövelheti a levél méretét.
- Ha egy olyan dokumentumot próbálunk megnyitni, amelyik sérült, a Word lefagyhat.
- Ha egy dokumentumról több mentést is készítünk ugyanabban a formában, majd MHTML formátumba exportáljuk, a fájl mérete minden egyes mentéssel exponenciálisan növekszik.
- Előfordulhat, hogy a word nem válaszol, ha a gépre Norton Antivirus van telepítve.
- A word nem válaszol a dokumentum újratördelésekor. Ezt az okozhatja, ha a dokumentum egy hosszú bekezdést tartalmaz.

Borsi Katalin
bobo@netacademia.net

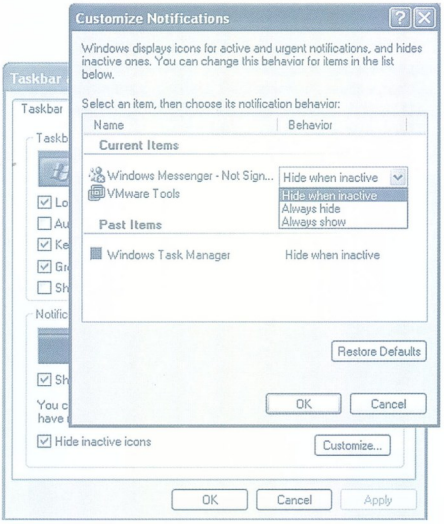
A cikkben szereplő URL-ek:
[1] http://msdownload.netacademia.net

Dupla KV

Windows XP értesítési területe

K: *Windows XP-ben hogyan lehet beállítani, mely ikonok jelenjenek meg a tálca jobb alsó sarkában levő értesítési területen (notification area?)*

V: El tudjuk tüntetni a tálcáról a nem használt ikonokat, illetve szabályozni tudjuk, hogy mikor látszódjanak és mikor ne. Az értesítési terület beállításait a tálcán *(Taskbar)* állva jobb klikk után a tulajdonságok menüt választva lehet megjeleníteni.



A tálca beállításai

A kép aljan is látható „Customize..." gombra kattintva feljön még egy ablak, itt állítható az értesítési területen megjelenő ikonok viselkedése. Három lehetőség közül lehet választani: vagy mindig látszódik az adott ikon, vagy egyáltalán nem, vagy pedig akkor látszik, ha van valami dolga *(és eltűnik, ha inaktív)*.

Windows 2000 jelszóváltoztatás

K: *Be lehet állítani valahogy a tartományban levő felhasználóknak, hogy csak akkor tudjanak jelszót változtatni, amikor arra az operációs rendszer figyelmezteti őket?*

V: Igen. Csoportos házirenden keresztül az összes felhasználóra vonatkozóan egyszerűen be lehet állítani azt, hogy maguktól ne tudjanak jelszót változtatni a felhasználók.

Az Active Directory Users and Computers snap-inben el kell navigálni ahhoz a szervezeti egységhez, amelyben a felhasználók vannak, de természetesen tartományi szinten is szabályozható a beállítás. Ha megvan a konténer, ahova rá szeretnénk húzni a beállítást, a következőt kell tenni:

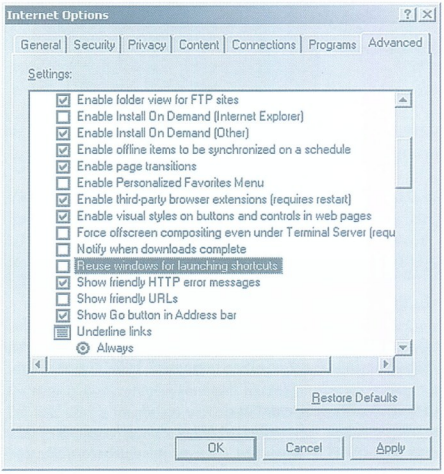
1. A konténeren jobb klikk – Properties – Group Policy tulajdonságlapon – Edit gombra kattintva megnyílik a házirend.
2. El kell navigálni a User Configuration – Administrative Templates – System – Logon/Logoff részére
3. A „Disable Change Password" házirendet kell bekapcsolni.

Ezzel a CTRL+ALT+DEL hatására feljövő ablakban a Change Password gomb inaktívvá válik, tehát a felhasználó itt nem tud jelszót módosítani. Egyetlen lehetősége marad: akkor változtasson jelszót, amikor az operációs rendszer erre figyelmezteti.

Internet Explorer parancsikon mindig új ablakban

K: *Megoldható-e, hogy az asztalon levő Internet Explorer parancsikon mindig új böngésző ablakban nyíljon meg? A környezet Windows XP és Internet Explorer 6-os.*

V: Igen. Az Internet Explorer Tools menüjében az Internet Options – Advanced ablakában meg kell keresni a „Reuse windows for launching shortcuts" beállítást – a Browsing szekcióban lesz. Az előtte levő jelölőnégyzetből ki kell venni a pipát.



Az Internet Explorer indítás beállítása

Az Exchange OWA Logoff gombja

K: *Az Exchange SP2 óta az OWA parancsikonok közt ott van a kilépéshez is egy ikon. Ha a felhasználó csak úgy becsukja az Explorer ablakot, vagy beír egy másik URL-t, akkor nem történik meg a kilépés. Van-e mód arra, hogy figyelmeztetést kapjon a felhasználó arról, hogy előbb lépjen ki, majd utána csukja be az Internt Exporert?*

V: Az Exchnage SP2-ben nemcsak a Kijelentkezés ikonja az újdonság, hanem egy hozzá kapcsolódó registry kulcs is, amellyel szabályozható, hogy ha a felhasználó nem szabályosan a Log Off vagy Kijelentkezés parancsikonra kattintva lép ki, figyelmezte-

tést kap. Ebből megtudja, hogy minden böngésző ablakot be kell csuknia az OWA használata után, mert csak így lehet biztos benne, hogy más nem fér hozzá az ő postaládájához.

Az extra figyelmeztetés megjelenítéséhez az Exchange kiszolgálón következő registry bejegyzést kell felvenni:

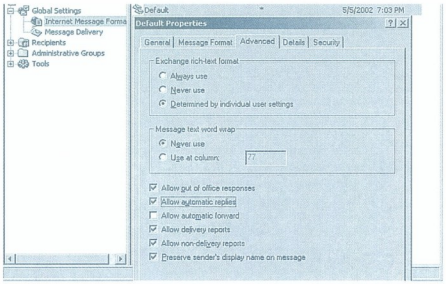
```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\
Services\MSExchnageWEB\OWA
Value:EnableLogoffWarning
Type: REG_DWORD
Data: 1
```

Ez a beállítás nem akadályozza meg a felhasználó kilépését, az ablak lecsukását, csupán figyelmezteti a felhasználót. Csak OK gomb van a figyelmeztető ablakon.

„Házon kívül" üzenetek házon kívülre

K: *Hol kell engedélyeznem, hogy az „Out of Office" automatikus válasz a kívülről jövő levelekre is válaszoljon? A környezet Exchange 2000 Service Pack 2-vel.*

V: Alapértelmezésben az „out of office" üzenetek küldése nincs engedélyezve az Internet felé. A beállítása elég eldugott helyen van: System Manager – Global Settings – Internet Message Format – Default policy – tulajdonságok – Advanced:



„Out of Office" üzenetek beállítása

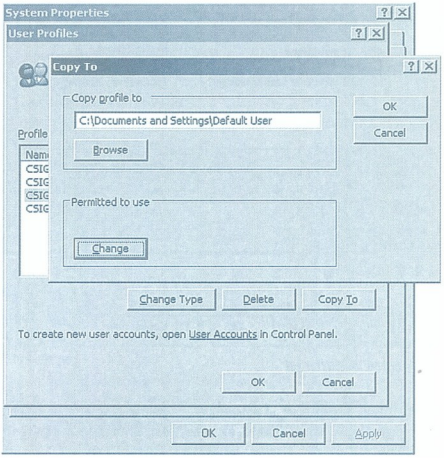
Ha sikerült eljutni a megfelelő ablakig, már csak egy pipára van szükség az „Allow out of office responses" jelölőnégyzetében.

Default Profile testreszabása

K: *Megpróbáltam egy default Profile-t csinálni egy gépen ahol több felhasználó is belép, úgy, hogy az Administrator felhasználóval beállítottam amiket akartam, majd az Administrator Profile könyvtárát bemásoltam a c:\Document and Setting\Default User könyvtár alá. Majdnem jó amit kaptam, de mégsem egészen. Mit rontottam el? Hogy kell ezt jól megcsinálni?*

V: Valóban majdnem jó a megoldás. A következő módon kell létrehozni „rendesen" az alapértelmezett profilt:

1. Be kell lépni a gépre Administratorként
2. Létre kell hozni egy felhasználót, legyen ez a Users csoport tagja
3. Be kell lépni a létrehozott felhasználó nevében
4. Be kell állítani mindent úgy, ahogy szükséges – nyomtatókat, az explorer beállításait, a hátteret, stb., vagyis mindent, ami kell.
5. Újra be kell lépni Administratorként
6. Mivel alapértelmezésben a rejtett fájlok nem láthatók, a profil másolásához be kell kapcsolni *(az Explorer – Tools – Folder Options beállítások közt)*, hogy ezek is láthatóak legyenek. Ehhez a Folder Options -View – Advanced Settings alatt a „Show hidden files and folders" beállítást kell bekapcsolni.
7. Most már lehet a profilt másolni, de nem egyszerűen fájlmásolással, hanem a Control Panel – System – User Profiles tulajdonságlapról! Itt a 4. pontban létrehozott felhasználó profilon állva a Copy To gombra kell kattintani. Ahogy a képen is látszik a Copy Profile To alatt kell megadni a könyvtárat, ahova a kiválaszott profilt másolni szeretnénk. Ebben az esetben a célkönyvtár a C:\Documents and Settings\Default User könyvtár.
8. A „Permitted to use" alatt pedig az Everyone-t kell megadni.



„Default Profile" előállítása

Az így létrehozott alapértelmezett profilt fogja megkapni minden új belépő az adott gépen.

Folytatás a 43-44. oldalról

kattintgatóhuszár „megfelelően" belerondít a DHCP beállításokba, cégszintű problémát tud okozni. Ismerek olyan céget, ahol egy kis DHCP tuningolás miatt negyven vidéki iroda két napig nem tudott dolgozni *(egy kis DNS-buhera, lease time megnövelése, click OK)*. És a DHCP még csak nem is összetett rendszer, s az sem igaz, hogy ne tudnánk pontosan, melyik módosításnak mi a következménye.

Most vizsgáljunk meg egy központosított és feltöltött Active Directoryt. Nyolcszázhuszonhárom felhasználó van benne, ízlésesen szervezeti egységekbe rendezve. Normális esetben majdnem mind a nyolcszázhuszonhárom fiók jogosultsága közel nullával egyenlő, hisz hagyományos felhasználókról van szó. Bármilyen pedáns rendszerre igaz az entrópia növekedésének elve: a sarokba szorított felhasználók idővel egyenletesen kitöltik a jogosultságteret. Előbb Piri kerül be a Backup Operators csoportba, hogy rá lehessen bízni a mentést, majd Rozi néni a Domain Adminsba, hogy nagy felbontással tudjon scannelni.

Ez utóbbi egyáltalán nem vicc! Van egy bizonyos scanner, mely alapfelhasználóknak maximum 92 DPI-s lapolvasást enged, rendszergazdáknak meg 300 DPI-t. *(Aki nem hiszi, írjon nekem, megadom a gyártó nevét és a pontos típust!)*. Egy gond van ezzel a működéssel: a jelenségről a gyártó nem tud, ők ilyet nem terveztek bele *(eredetileg nem is volt hozzá NT-s meghajtó)*. Így a tudomány mai állása szerint egyetlen módon tudjuk gyalogok számára biztosítani a normális működést: „ideiglenesen" betesszük őket a Domain Admins csoportba. Ettől persze olyan műveletek is megjelennek számukra a könyvelőrendszerben, ami addig soha, és olyan lekérdezéseket futtathatnak a cégadatbázison, amit a vezérigazgató is megirigyelne. És előttünk áll Rozi néni, a hacker, aki semmit sem tett azért, hogy létfontosságú, titkos adatokhoz jusson. IPSec? EFS? Rozi néni ezeket mind öntudatlanul cselezte ki! Rozi néni egy kétlábon járó side effect *(mellékhatás)*.

Idővel egy csomó felhasználó rendelkezni fog egy halom olyan jogosultsággal, amiről már senki nem tudja, hogy miért adták ki, de senki nem meri visszavonni sem. A gyakorlat azt mutatja, hogy a „tesztelési céllal" kiadott jogok örökre úgy maradnak, a felhasználók pedig egyre erősebb csoportokba diffundálnak.

Mit lehet itt tenni? Utólag szinte lehetetlen visszaverni a józanszabadcsapatokat. Valahogy meg kellene tudni akadályozni, hogy a folyamat beinduljon.

Glokalizáció

Az egyik megoldási módszer a véletlen mellékhatások esélyének csökkentése, vagyis - mondjuk ki - az általános központosítás „megfúrása", olyan szigetek kialakítása, ahova más rendszerek szuperjúzerei sem jutnak be automatikusan. A glokalizáció fogalma általánosan azt jelenti, hogy globalizáció ide vagy oda, a részrendszerek autonómiáját érdemes lehet meghagyni, mert többet érnek, ha bizonyos függetlenséggel rendelkeznek. A fenti fürt-sziget csak veszítene azzal, ha beleolvadna a közös rendszerbe. Ezzel nem azt mondom, hogy ne központosítsunk! Egységesítsük, amit érdemes, de csak azt!

Feudalizmus

Mit egységesítsünk? A nyilvántartásokat igen *(címtár, adatbázisok)*, a jogosultságkezelés központosítását azonban már fontoljuk meg. Vegyünk egy történelmi példát, Magyarországot. Egységesítsük az állampolgárok nyilvántartását *(központi címtár)* és azonosítását *(autentikáció)*, de hagyjuk a falvak önkormányzataira annak eldöntését, hogy ki kap szocpolt, és ki nem.

Utoljára a feudalizmusban léteztek olyan globális csoportok *(báró, gróf stb.)*, amibe ha bekerült valaki, minden ajtó megnyílt előtte. A globális csoportok hatékonyak a nyilvántartás szempontjából, de ügyetlen használatuk feudalizmushoz vezet. A nemesi címet adományozzák, de vissza nem veszik. Az egyenlősdi ismételt eléréséhez forradalom kell!

Itt az ideje, hogy visszakanyarodjunk a tech.edhez. Don Box előadása nagyon röviden arról szólt, hogy sajnos napjainkban ismét fel kell találni a spanyol viaszt, ismét olyan alapszolgáltatásokat kell kifejleszteni, amelyekről már mindenki azt hitte, többé nem kell velük vacakolni. Két ilyet említett: a protokollokat és a tranzakciókat.

Protokollok

A harminc éve velünk élő TCP/IP-t arra találták ki, hogy ügyesen elrejtse előlünk a hálózat bizonytalanságait, és nekünk, felhasználóknak hibátlanul működő csatornát nyújtson, amibe csak bele kell kiabálni, s a cső végén hibátlanul bukkan fel eredeti üzenetünk. Egy hagyományos ügyfél-kiszolgáló alkalmazás számára ez maga a kánaán: reggel nyolckor megnyit egy TCP csatornát az SQL Server felé, és többé nem kell aggódnia az adatátvitel miatt. Csakhogy egy webalkalmazás nem így működik. TCP csatornák jönnek-mennek, bomlanak-épülnek, így szegény alkalmazásnak saját kezébe kell venni a protokollkezelést: el kell hárítania a csomagok elveszítéséből és kaotikus sorrendjéből eredő problémákat. Még bonyolultabb protokollépítésre lehet szükség webfarmok esetén, ahol még csak esély sincs arra, hogy a feladatot a TCP/IP elvégzi helyettünk. Bizony, ott tartunk, hogy az alkalmazásnak kell figyelnie a helyes adatmozgásra:

1. Vajon bejelentkezett már a felhasználó?
2. Vajon megkapta már az alapadatokat? Vagy a semmiből válogat?

Még olyan helyzetekre is fel kell készülnie, ami egy hagyományos alkalmazásnál nem fordulhat elő:

1. A beérkezett rendelés valós adatokat tartalmaz, vagy a felhasználó belehekkelt az URL-be?
2. Ha nyomogatja a Frissítés gombot, ez hány rendelésnek számít?
3. Hányadszor küldi el a gonosz ugyanazt a rendelést különböző random paraméterekkel?

Vállalati hálózatunkra visszatérve: ha nem szeretnénk áldozatul esni az előre nem várt helyzeteknek, vegyük őket számba, alkossunk protokollt az esetek lekezelésére! Minél összetettebb a rendszer, annál fontosabb, hogy kész receptünk legyen, mi a teendő „nem várt" esetekben.

A Microsoftnak van egy átfogó, nyolcszáz oldalas „protokollja" vállalati hálózatok üzemeltetésére. Úgy hívják: Microsoft Operations Framework *(MOF)*. Részt vettünk a magyarításában, ezért tudom: ennél szárazabb olvasmányt elképzelni is nehéz. De nem is ponyvaregényként kell használni, hanem Bibliaként. Fel kell ütni a megfelelő versnél, és annak megfelelően cselekedni. Rengeteg tervem közt szerepel, hogy egyszer, ha majd nagyon ráérek, írok egy cikket a MOF-ról...

Felhasználói tranzakciók

Tíz éve mondogatják nekünk, hogy az adatbáziskezelés elérte végcélját, mivel a tranzakciókezelés miatt nem fordulhat elő olyan módosítás, mely önellentmondásba viszi az adatokat. Tíz éve mondom, hogy ez igaz, de ha kidugjuk a fejünket a homokból, kiderül, hogy az adatbázisnak nemcsak önmagával, hanem az általa leképezett világgal is ellenmondásmentesnek kell lennie. Sokra megyünk egy olyan adatbázissal, mely a furfangos világ miatt más dimenzióban él, mint mi!

(Kiváló példa: egy könyvtári adatbázis automatikusan figyelmeztető státuszba teszi azokat a tagokat, akik nem vitték vissza idejében a kikölcsönzött könyveket. Egyesek szabályosan tesznek eleget a felszólításnak, visszaviszik a könyvet, kifizetik a büntetést, és lekerül róluk az „átok". Mások suttyomban visszacsempészik a könyvet a polcra, s

ezzel elintézettnek veszik az ügyet. Az adatbázisban a tag státusza érintetlen marad. Melyik valóság a valóságosabb? Az adatbázisé, vagy a könyvespolcé, melyen ott áll a kintlévőnek hitt könyv?)
Ha körülnézünk a világban, észrevehetjük, hogy tranzakciók vesznek körül minket. Lefoglalunk egy asztalt az étteremben *(BEGIN TRAN)*, majd elmegyünk vacsorázni *(COMMIT)*. Félretetetünk egy mozijegyet *(BEGIN TRAN)*, de mégsem megyünk érte *(ROLLBACK)*. Elhatározzuk, hogy vegetáriánusok leszünk *(BEGIN TRAN)*, három hónapig bírjuk, majd lecsúszik az első karika kolbász *(ROLLBACK)*. Ha jobban megfigyeljük, ezek nem is igazi tranzakciók.

- Nem kell elválasztanunk *(izolálnunk)* a többi ember folyamatától, nem baj, ha mások is tudják. Három hónapig valódi vegetáriánusok vagyunk.
- Nem baj, ha esetleg hosszúra nyúlik a folyamat. Ha vissza is vonjuk, nem biztos, hogy minden hatása megszűnik. ***(Nem pótolható a három elvesztegetett hónap, amíg botor módon nem ettünk húst!)***
- Bármikor, további következmények nélkül meggondolhatjuk magunkat *(mától mégsem vagyok vegetáriánus)*.

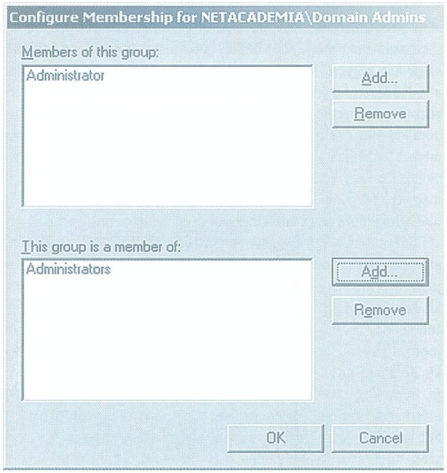
Ez nem tranzakció, nem is UNDO, hanem kétállapotú kapcsolók át- és visszabillentése, miközben az élet megy tovább. Néhány kapcsoló magától visszabillen *(a félretett mozijegy nem a mienk, ha nem megyünk érte)*, másokat mi magunk billentünk vissza. Az informatikai rendszerek kezelése is billenőkapcsolók átállításából áll. Bekerüljön-e Rozi néni a Domain Adminsba? Igen? Nem? Végül is betettük a Domain Adminsba. Egy hétig ezzel a jogosultsággal tett-vett, és azt szeretnénk, hogy idővel magától szűnjön meg a csoporttagsága. Ne kelljen gondolnunk rá!
Képzeljünk magunk elé egy olyan rendszert, mely mindig megjegyzi, mit billentettünk át, és egy naplóból pontosan vissza tudunk keresni, esetleg meg nem történtté tenni néhány eseményt. Mikor állt elő a hiba? Tegnap délben? Mit csináltunk közvetlenül előtte? GPO-t faragtunk? Kell ez nekünk? Nem!
Miért mondják, hogy utópista fantazmagória? Nem ugyanezt csináljuk ma is, puszta kézzel, kockás papíron jegyzetelve, ki-ki hagyva egy-egy módosítást a naplóból *(mert most nem fontos, hanem sürgős)*? Jobban esik szalagos mentésről visszaállni? Milyen állapotba is?
Gépesítsük, ami gépesíthető! Szerencsére mind a protokoll-, mind az állapotkezelés automatizálásában vannak eredményei a Microsoftnak.

Automatizált protokollok

A különböző bonyolult folyamatok dokumentálásában érdekes módon nem a Microsoft járt az élen, hanem a NetIQ nevű cég. Az ő IQ-juk nem más, mint rengeteg szakember tapasztalatának összegyűjtése egy önjáró rendszerfelügyeleti keretrendszerbe. A NetIQ szoftverét a Microsoft licenceli, neve: Microsoft Operations Manager *(MOM)*. A MOM egyik feladata a protokollkezelés. Ha ez és ez történt, csináljuk azt. Ha háromszor előállt ezmegaz, legyen helyette ingyombingyom. Ha már x órája nincs bakfitty, küldjünk egy pitypalattyot. Ha valamelyik rendszer A-t mondott, mondjon B-t is. Ha elfogyott a gezemice, riasszuk a rendszergazdák főnökét. Csupa olyan feladat, melyet mi magunk is így végeznénk – ha eszünkbe jutna. Hányszor fogyott ki alólunk a szabad lemezterület, csak mert senki sem figyelt rá? A MOM-nak eszébe jut! Hihetetlen mennyiségű *(több száz)* előre elkészített rendszergazdai feladat áll benne készenlétben, nekünk csak be kell kapcsolni.

Állapotvisszakapcsolás

Akár hiszik, akár nem, az Active Directoryban bizonyos funkciókra már ma is van UNDO, igaz, válogatni nem nagyon lehet sem abban, mit csináljon vissza, sem abban, hogy mikor. De a Microsoft igyekezete figyelemre méltó. Még megérjük, hogy legvadabb képzelgéseink is testet, illetve szoftvert öltenek!
Az auto-undo elvégzésére a csoportos házirendet *(Group Policy)*, annak is biztonsági leágazását nevezték ki. Bizonyos objektumok jogosultságait, és Active Directory csoportok tagsági körét lehet vele beállítani, amelytől el tudunk térni, de az AD minduntalan visszatér. Mitha egyenesen a scanneres példára készítették volna! Csak betesszük Rozi nénit a Domain Adminsba, s majd a házirend gondoskodik arról, hogy idővel kitakarodjon onnan. Nincs többé ottfelejtés! Lássuk, hogyan működik!
Az Active Directory Users and Computerssel megnyitjuk a Default Domain Controllers házirendet. Ennek gépi szekciójában *(computer configuration)* a Windows Settings->Security Settings alatt találjuk a Restricted Groups (kötött csoportok) beállítást. Jobbklatty, Add group, Domain Admins. A létrejövő objektumot megnyitva ezt látjuk:



Kötött csoportok használatával védekezhetünk saját elfelejtett kicsapongásaink ellen

Az ablak felső részébe vegyük be azokat a felhasználókat, akiket minden körülmények közt bent szeretnénk látni a csoportban. Az alsó részben pedig azokat a csoportokat soroljuk fel, melyeknek a Domain Admins mindenképpen tagja. A fenti ábra egy lehetséges megvalósítást mutat.
Lassan be kell fejeznem a „bevezetőt", így házi feladatba adom Önöknek, hogy merengjenek el a kötött csoportok alatt található System Services, a File System és a Registry beállításokon. Ezek is automatikus visszaállításra valók!
Ezzel zárom soraimat, üdvözlet Barcelonából!

marcellf@netacademia.net
MZ/X

A cikkben szereplő URL-ek:
[1] http://barcelona.netacademia.net

Előfizetési szelvény

Címzett: NetAcademia Kft.
Faxszám: (1) 261-7145

fax

Tisztelt Olvasónk!
Lapunk hírlapárusi forgalomba nem kerül, ezért ha kíváncsi megkezdett sorozataink folytatására, kérjük töltse ki és juttassa el hozzánk az alábbi megrendelőlapot. Előfizetőink kedvezményesen vehetnek részt konferenciáinkon, tanfolyamainkon illetve egyéb rendezvényeinken.
Kérjük töltse ki ezt az előfizetési szelvényt és faxolja el az (1) 261-7145-ös faxszámra.

Új előfizetési AKCIÓ!

A 2002. április 1-től indult új előfizetési akciónkban minden új előfizetőnk aki 2002. július 30-ig egy évre előfizet a tech.net magazinra, ajándékul megkapja tőlünk a 2002. januári, februári és márciusi számokat. Fizessen elő egy évre most, hogy küldhessük Önnek az ajándék magazinokat!

Az akció határideje: 2002. július 30.

http://technet.netacademia.net/subs • e-mail: terjesztes@netacademia.net • fax: (1) 261-7145

Előfizetem a **tech.net** magazint: ...példányban egy évre (14.784 Ft)
...példányban fél évre (7.392 Ft)
...pld.-ban.NET akcióval ***(12+3 szám)*** (14.784 Ft)

Az előfizetés kezdete:...../...../.....

Előfizető neve:. . . . . .

Cég neve: . . . . . .

Cím: ☐☐☐☐ . . . . . .

E-mail cím: . . . . . .

Telefon: . . . . . .

Fax:. . . . . .

Fizetés módja: ☐ csekken *(postán küldjük)* ☐ átutalással

Kelt:...../...../..... Aláírás:.....

Amennyiben a számlázási cím nem egyezik meg a szállítási címmel, kérjük az alábbi részt is töltse ki!

Számlázási cím: **Szállítási cím:**

....

....

tech.net working with windows