

.NET  
WINDOWS 2000 • ACTIVE DIRECTORY  
SQL SERVER 2000 • TRANSACT SQL  
SZABVÁNYOK • RFC 1597  
DUPLA KV  
OFFICE 2000 • SERVER EXTENSIONS  
DEVELOPER • ADSI

# tech.net

A MICROSOFT MAGYARORSZÁG SZAKMAI MAGAZINJA

|   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |



0 0 1 1 1 0 0 1 0 0

# 2000.10.

# KÖSZÖNTŐ

## Fülöp Miklós főszerkesztő-helyettes



### Kedves Olvasók!

Elérkeztünk a Tech.Net magazin második számához. Az első számba bűt szárvashibák (nevezzük őket *gyermekbetegségeknek*...) után nagyon kíváncsiak voltunk az olvasói visszajelzésekre. Hisszük, hogy lapunk hiányt pótol, de azt nem hit-tük volna, hogy ennyire nagy hatás-sal lesz a magyar Microsoft-felhasz-nálók informatikus világára.

Magazinunk első számához az juttot hozzá, akinek a Microsoft Magyaror-szág ingyen adott: cégek, partnerek postán, vagy rendezvé-nyeken kapták kézhez lapunkat. Az újsághoz tartozó levele-zési lista azonban nyilvános, mindenki által elérhető, így tulajdonképpen az újságot megelőzte a híre.

Itt érkezünk el ahhoz a ponthoz, amire valójában (még) nem készültünk: az óriási érdeklődés mellett mindenki az előfizetési lehetőségekről kérdezősködött, mi pedig csak annyit mondhattunk: elnézését kérünk Uram és Hölgyeim, ez a dolog valójában még nincsen kitalálva! Bevallhatjuk őszintén, nem számítottunk már az első hónapban ekkora érdeklődésre, tulajdonképpen az első szám megjelenése pilan-tatában fogalmunk sem volt arról, hogy ki, hogyan és mi-kor fogja terjeszteni az újságot.

Azonnal nekifogtunk kidolgozni az előfizetési konstrukciókat (és az *infrastruktúrát*), mert már az a veszély fenyegetett, hogy fénymászó kalózpéldányok kezdenek el terjedni, mint valami forradalmi lapnál... nos, kedves olvasók, az előfizetés mostantól működik. Aki szeretné, hogy lapunkat minden hó-napban biztosan megtalálja a postaládájában, nincs más dol-ga, mint ellátogatni a <http://tech.net.netacademia.net> cím-re és ott megadni előfizetői adatait (*ha úgy kívánalmesebb, szívesen várjuk az e-maileket és faxokat is a túldolgaló imp-resszumban található címre és telefonszámra is*).

Második szám, mondtam, holott ez valójában nem igaz: a Microsoft Magyarország .net Enterprise bejelentésére idő-zést megjelent első különszámunk is, benne a .net kiszol-gálócsalád (a *volt BackOffice*) tagjainak ismertetésével – ezúton is köszönjük a vezető informatikai cégek szakembereinek támogatását. A különszám szerkesztőségünkben megvásárolható, sőt, aki lapunkat egy évre előfizeti, az eh-zhez – és a következő különszámokhoz is – jelentős kedvez-ménnyel juthat hozzá.

Még egy utolsó gondolat az előfizetésről: olvasóink kérdez-ték, érdemes-e előfizetni, van-e jövője ennek a magazin-nak? Nos, mi hisszük, hogy van. Az informatikus szakem-bernek éjjel-nappal, szinte örült tempóban kell követnie a változásokat. Ha valaki két hétre szabadságra megy (*s mint tudjuk, a mobiltelefon, a laptop nem tűri jól a tengerparti homokot :-)*), visszatértekor szinte tanulhatja újra a mes-terségét. Mi az újdonságok bemutatására törekszünk, és amíg újdonságok lesznek, addig reményeink szerint a *tech.net* magazin is talpon marad majd. Nyilvánvaló, hogy az érintett témák és azok mélysége meghatározza, egy-

szerszint korlátozza az olvasói kör terjedését, de azt sze-retnénk, hogy aki ilyen típusú információra „éhes”, az meg-kapja tudását a havi betevőt.

Egy másik olvasói megjegyzésre reagálva: nem célunk az, hogy eladjuk ezeket a termékeket, hanem az, hogy meg-könnyítsük azok professzionális használatát. Természetesen valamilyen szinten hiszünk az itt bemutatott technológiák-ban, különben nem lennénk itt, de az is lehet, hogy a tu-dásvágy, az újdonságok iránti érdeklődés hajt minket – és reméljük, olvasóinkat is.

Célunk az, hogy bemutassuk az új lehetőségeket, lehetőleg szakmai szemmel, ezért hiányszik most – és nagy valószínű-ség szerint a jövőben is – lapunkból a játékleírás, a CD-mel-léklet. Nincs szándékunkban felvenni a versenyt a témával régebb óta, részletesen foglalkozó más lapokkal, és úgy érez-zük, erre tulajdonképpen nincs is szükség. Ha mégis valami olyan különleges, egyedül dologhoz jutunk hozzá, mint példá-ul egy nyilvános béta verzió, akkor előfordulhat, hogy meg-lepetésszerűen az újsághoz postázunk egy-egy fényes koron-got is, de egyelőre nem tervezzük hasonló akciót. Ha olyan szoftvert jut a birtokunkba, ami széleskörű érdeklődésre tart-hat számot, és a méretbeli és jogi körülmények azt lehetővé teszik, inkább közzétesszük az újság készülő weblapján, ahonnan bárki letöltheti, akinek szüksége van rá.

A tartalomról: a .net bejelentése ha lehet, még jobban fel-kavarja az informatika amúgy sem álló vizet. E számban egy-két cikk erejéig foglalkozunk a .net filozófiájával, a kö-vetkező számtól pedig esélyünk sincs arra, hogy ne erről beszéljünk: legyen a téma egykori BackOffice termék – most a .net Enterprise kiszolgálócsalád tagja; legyen fejlesztői környezet – mostantól Visual Studio.Net a neve; legyen az operációs rendszer, legyen Active Directory, legyen az új technológia – Windows 2000, Active Directory, XML, SOAP, WebDAV, mind-mind a .net-világ alapját képezi. Összefog-lalva a .net lényegét: bármit, bárhol, bármivel!

Hogy ez a három varázsszó tulajdonképpen mit takar, arra a következő néhány hónapban fény derül majd. Mi a ma-gunk részéről annyit tehetünk, hogy ismertetjük ezeket az építőköveket. Jelen számunkban mesélünk az Active Directory egyszerű programozásáról, az ADSI-ről, a .net egyik felhasználási eszközeiről (*surprise!*), az Office-ről, és az Office Server Extensions-ről, elárulunk néhány titkot az új SQL Server 2000-ről és még sok-sok másról.

Jó böngészést kívánok!

Fülöp Miklós  
mick@netacademia.net  
MCSE, MCT

Microsoft



# tech.net

A Microsoft Magyarország Szakmai Magazinja

Szerkesztőség

Főszerkesztő: **Fóti Marcell**

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

Főszerkesztő-helyettes: **Fülep Miklós**

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

Szerkesztőség címe:

1105 Budapest, Ihász utca 13.

Tel.: 263-2732

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

Nyilvános levelezési lista:

[technet@lyris.netacademia.net](mailto:technet@lyris.netacademia.net)

Kiadja a Microsoft Magyarország

1031 Budapest, Graphisoft park 3.

Tel.: 437-2800

A kiadásért felel:

**Arany Tóth László**

[y-laarto@microsoft.com](mailto:y-laarto@microsoft.com)

Terjeszti a NetAcademia Kft.

Terjesztési, előfizetési információ:

Tel.: 263-2732

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

Megjelenik havonta, ára 899 Ft

Előfizethető megrendelőlevélben a

szerkesztőségéknél:

1105 Budapest, Ihász utca 13.

Fax: 261-7245

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

Hirdetésfelvétel:

**Bársonykalapács Marketing**

Felelős: **Udvarev Rita**

Tel./Fax: 214-0923

[velvethammer@ahol.com](mailto:velvethammer@ahol.com)

1027 Budapest, Fő utca 67. V. 1.

Grafikai tervezés, kivitelezés,

nyomdai előkészítés:

**Bársonykalapács Marketing**

Művészeti vezető: **Balogh Zoltán**

Nyomda:

**Partner's 2000 Kft.**

1124 Budapest, Sion lépcső 7.

Felelős nyomdász: **Galambos Sándor**

ISSN 1586-5185



## Windows 2000

.NET 3. oldal

Active Directory 8. oldal



## BackOffice

SQL Server 2000 Transact SQL 13. oldal



## Szabványok

RFC 1597 17. oldal



## Hírek

20. oldal



## Biztonság

IIS és az NTSF 21. oldal



## Developer

ADSI 25. oldal



## Dupla KV

Kérdések és válaszok 32. oldal



## Business Internet

IIS 5.0 33. oldal



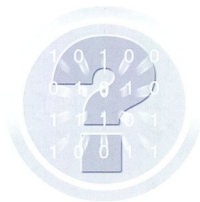
## Office 2000

Server extensions 40. oldal



## Bill Gates mondja...

A .NET technológiáról 45. oldal





Microsoft .NET: Az Internet új generációja

# Microsoft .net

Microsoft tanulmány - 2000. június

Bevezetés: a forradalmi üzlet

A számítástechnika a forradalmak ipara. Alig 20 éve még mindenki a mainframe-ről beszélt. Keveseknek adatot meg a számítógépezés lehetősége, és nekik is többnyire egy cég számítástechnikai osztályán. A PC és a grafikus felhasználói felület ennek véget vetett: demokratizálta a számítástechnikát, hogy emberek milliói számára legyen elérhető, és valódi tömegcikké változtatta a számítógépet. A vállalatok felismerték, hogy a PC-s hálózatok és a PC-alapú kiszolgálók átalakíthatják üzletvitelüket, ugyanakkor a fogyasztók körében a PC rövid idő alatt házi szórakoztató eszközzé nőtte ki magát. Aztán megjelent az Internet. Forradalmasította a kommunikációt, megteremtette az információ és a szórakoztatás új és gazdag forrását, és megtoldotta egy e betűvel a kereskedelmet. Ma közel 300 millióan használják az Internetet. Az International Data Corp. szerint az Internet ebben az évben több mint 250 milliárd dolláros üzleti forgalmat bonyolít le.

Mindezek ellenére még rengeteg a fejlődési lehetőség. A mai Internet nagymértékben a régi mainframe-modell tükörképe. A sávszélesség bősége ellenére az adatokat most is központi adatbázisokban tárolják, amelyekhez a hozzáférést „kapuőrök” ellenőrzik. A korábbi időosztásos módszerhez hasonlóan a felhasználóknak minden művelet elvégzését a webkiszolgálóra kell bízniuk. A webhelyek elkülönült szigetek, amelyek a felhasználó nevében értelmes módon nem tudnak kapcsolatba lépni egymással. A web jelenleg alig több, mint olyan, egymástól független oldalak halma, amelyek nem az adatot, hanem annak HTML „képét” tartalmazzák. *(A legtöbb webhely esetében ma még mindkét feltétel teljesítése túl nagy követelményt jelentene.)* Sok tekintetben a böngészők sem többek buta termináloknál: az adatok keresése könnyű, ellentétben a szerkesztéssel, elemzéssel és manipulációval *(azaz minden olyan dologgal, amire egy tudásalapú rendszerben szükség lehet).* A teszteszabás eredményeként a meglátogatott webhelyek által kért személyes adatok egymással átfedésben vannak, ráadásul az ellenőrzésükről is le kell mondanunk. Ahelyett, hogy a technológia alkalmazkodna hozzánk, nekünk kell alkalmazkodnunk a technológiához.

Több PC vagy hordozható eszköz használatakor ezek a problémák fokozottabban jelentkeznek. Az online információk, elektronikus levelek, offline fájlok és más adatok elérését megnehezíti a – *gyakran inkompatibilis* – csatlófelületek végtelen változatossága, az adathozzáférés többféle szintje és a szinkronizáció korlátozott lehetősége *(ami a hordozható eszköz és a PC fizikai összekapcsolását jelenti)*. Az online adatokat hiányosan definiálják, ami nagymértékben korlátozza a felhasználásukat. A szükségleteinkhez alkalmazko-

dó testre szabott „személyi információs tér” koncepciója egyelőre még alom.

A webhelyek tervezésére, tesztelésére és kivitelezésére szolgáló fejlesztői eszközök nem megfelelőek. Sokszor nagyobb hangsúlyt fektetnek a külsőre, mint a használhatóságra, és egyik sincs tekintettel a szoftvereknek a tervezést, fejlesztést, kivitelezést, üzembehelyezést és karbantartást is átfogó teljes életciklusára. Jelenleg nincs lehetőség arra, hogy a PC-re írt kódot különféle eszközökön változtatás nélkül használhassuk.

A vállalati felhasználóknak további kihívásokkal is szembe kell nézniük. Bár a kisebb kiszolgálókból álló fürtök az egy pontos hibák kiküszöbölésével növelték a megbízhatóságot, a rendszerfelületek bonyolultabbá váltak. A mai többszintű és többfunkciós webhelyek teljesítményének mérése, kapacitásának tervezése, üzemeltetése nem egyszerű feladat. Az új e-kereskedelmi és a hagyományos üzleti rendszerek megfeleltethetősége vagy együttműködése ritka. A fogyasztók és partnerek számára a vállalati információs rendszer megfelelő használatát biztosító, a túzfalon keresztül biztonságosan működő rendszerek tervezése olyan nehéz, hogy sok vállalat inkább a költségesebb megkészserezést választja. Tényleg olyan jó lesz az Internet, amilyenek most látjuk? A web további fejlődését senki sem kérdőjelezi meg, de ahhoz, hogy ez valóban hasznos legyen a fejlesztők, a vállalatok és a vásárlók számára, egy új, radikális vízió is szükséges. A Microsoft célja ennek a vízióknak és a megvalósításához szükséges technológiának a megteremtése.

## Microsoft.NET: a böngészők és a .COM világán túl

A Microsoft olyan fejlett szoftvergenerációt hoz létre, amely a számítástechnikát és a kommunikációt forradalmi módon egyesíti, a fejlesztők számára pedig eszközöket nyújt a web és a számítástechnika más területeinek átalakítására. A Microsoft.NET az első olyan kezdeményezés, amely a fejlesztők, vállalatok és vevők számára biztosítja, hogy a saját elképzeléseik szerint hasznosíthatóak a számítástechnika lehetőségeit. Lehetővé teszi valódi elosztott webszolgáltatások készítését, amelyek a kiegészítő szolgáltatások együttműködésével a .COM rendszereknél lényegesen jobb kiszolgálást valósítanak meg. Az Internet következő generációjában a Microsoft.NET meghatározó szerepet játszik, mert az adatokat bármikor, bárhol és bármely eszközön elérhetővé teszi.

A Microsoft.NET a figyelmet az egyedi webhelyekről és Internetre kapcsolt eszközökről a számítógépek, eszközök és szolgáltatások online együtteseire irányítja, amelyek az együttműködésük révén az eddigieknél sokoldalúbb megoldásokat nyújtanak. A felhasználóknak múlik, hogy milyen adatok, hogyan és mikor jutnak el hozzájuk. E széleskörű együttműködés révén megszűnnek az elszigetelt szolgáltatások, ahol az integrációt maguk a felhasználók végezték el. A vállalatok képesek lesznek oly módon kínálni termékeiket és szolgáltatásaikat, hogy a vásárlók transzparenensen beilleszthessék ezeket saját elektronikus rendszerükbe. Ez a vízió tovább növeli a PC-k 1980-as évekbeli megjelenésével kialakult személyi szabadságot.



A Microsoft.NET elősegíti az Internet átalakulását, ami a HTML alapú oldalak kiterjesztését jelenti programozható, XML alapú formába. Az XML széles körben elfogadott ipari szabvány, melyet a World Wide Web Consortium hozott létre. Nem kizárólagosan a Microsoft nevéhez fűződő technológia, bár a kidolgozásában jelentős szerepet vállalt a cég. Az XML segítségével az adat teljesen elválasztható a megjelenéstől. Az XML az Internet új generációjának kulcsfontosságú eleme: az adatok „felszabadításával” megkönnyíti a weblapok tervezését, programozását és javítását; hatékonyabbá teszi az adatok elosztását; lehetővé teszi webhelyek, illetve egymással kapcsolatot teremtő képes webszolgáltatások együttműködését.

A Microsoft.NET-et a következő elemek alkotják:

☞ **Microsoft.NET platform** – Magában foglalja a szolgáltatások új generációjának készítésére és üzemeltetésére alkalmas .NET infrastruktúrát és eszközöket; sokoldalú ügyfélprogramok készítését elősegítő .NET felhatalozott felületet; a nagymértékben elosztott megaszolgáltatások új generációját, a .NET szolgáltatásmódulatokat; és az internetes eszközök új nemzedékét támogató .NET szoftvereket.

☞ **Microsoft.NET termékek és szolgáltatások**

Tartalmazza a Windows.NET-et, amelybe foglalta a legfontosabb szolgáltatásmódulatokat: az MSN.NET-et; egyéni előfizetési szolgáltatásokat, az Office.NET-et; a Visual Studio.NET-et, és a bCentral for .NET-et.

☞ **Kívülről érkező .NET szolgáltatásai** – Számos partnernek és fejlesztőnek lesz lehetősége a .NET platformra épülő vállalati és vertikálisan integrálható szolgáltatások készítésére.

A Microsoft.NET a számítástechnikában és a kommunikációban jelentős elmozdulást eredményez az egyirányú webtől egy sokoldalú, együttműködésre képes, interaktív környezet felé. A Microsoft.NET új, fejlett szoftverek segítségével aknázza ki az alkalmazások, szolgáltatások és eszközök együttesét a testre szabott digitális élmény megteremtéséhez, amely folyamatosan és automatikusan alkalmazkodik a felhasználók, családok, otthonok, munkahelyek igényeihez. Ez egy integrált szolgáltatásként működő új szoftvergenerációt jelent, amely az Internet korszakában megkönnyíti életünk és munkánk szervezését.

Mit jelent ez a különböző szereplők számára?

☞ **Fogyasztók:** a szolgáltatások integrációjából fakadó egyszerűség. Dokumentumok készítésének, szerkesztésének és böngészésének egyszerűsítése. Fájlok, munkanyagok és egyéb média online és offline elérése. Az eszközök által nyújtott holisztikus élmény. Teljeskörű testre szabás-nincs menedzsment. Ez például azt jelenti, hogy a megváltozott adatok – történjen az adatbevitel akár egy PC-ről vagy egy hitelkártyáról – azonnal és automatikusan elérhetőek lesznek bárhol, ahol éppen szükség van rájuk.

☞ **Tudásalapú rendszerek és vállalatok:** dokumentumok készítésének, szerkesztésének és böngészésének egyszerűsítése. Sokoldalú, irányított kommunikáció. A hordozható eszközök nem követelnek többlettudást a fel-

használóktól. Hatékony adatkezelési és e-kereskedelmi eszközök, amelyek transzparensen váltogatnak a belső és Internetalapú szolgáltatások között, valamint támogatják a dinamikus kereskedelmi kapcsolatok új korszakának kialakulását.

☞ **Független szoftverfejlesztők:** lehetőség az Internet-korszaknak megfelelő új, fejlett szolgáltatások készítésére, amelyek az eszköztől vagy a nyelvtől függetlenül képesek az adatok helyi vagy távoli, automatikus elérésére és módosítására anélkül, hogy a kódot minden egyes környezethez újra kellene írni. Minden, ami az Interneten található, a szolgáltatások ezen új generációjának építőeleme lehet, ugyanakkor minden alkalmazás működhet internetes szolgáltatásként is.

A Microsoft.NET víziója szerint a fogyasztók, vállalatok, szoftverfejlesztők és a teljes ipar hatékonysága növekszik, az Internet – béklyóitól megszabadulva – kiterjedhet, és végre olyan lesz, amilyennek mi, felhasználók szeretnénk.

### A Microsoft.NET platform: az Internet új generációjának építése

Az XML és az Internet-protokollok integrációjára épülő Microsoft.NET platform olyan forradalmi modell, amely alkalmas egy új és fejlett szoftvergeneráció kifejlesztésére. A programozási modellek korábban egyetlen rendszerre összpontosítottak, és a más rendszerekkel való kapcsolatokat is úgy próbálták ábrázolni, hogy azok helyinek látszának. A Microsoft.NET-et kifejezetten arra tervezték, hogy az Internet erőforrásait integrálhatók legyenek egyetlen megoldással. Az integrációnak ez a fajtája ma bonyolult és költséges, de a Microsoft.NET ezt a szoftverfejlesztés lényegi elemévé teszi.

A lazán kapcsolt, XML alapú Microsoft.NET programozási modell bevezeti az XML alapú webszolgáltatások készítésének koncepcióját. Míg a mai barkácsolt webhelyek további jelentős fejlesztés nélkül képtelenek társikkal együttműködni, a Microsoft.NET programozási modellje erre könnyedén képes webhelyek és -szolgáltatások készítését teszi lehetővé. A Microsoft.NET úgy fogja elősegíteni az Internet következő generációjának kifejlődését, ahogy az egymást felváltani képes alkotóelemek megjelenése felgyorsította az ipari forradalmat.

De mindez nem jöhet létre a sok-sok partner, illetve független és vállalati fejlesztő nélkül, akik közreműködtek a mai számítástechnikai ipar kialakításában; vagy ahogy Alexander Graham Bell fogalmazott: „A nagy felfedezések és fejlesztések mögött mindig számos elme együttműködése áll.” A következő három évben a Microsoft 2 milliárd dollárt fordít arra, hogy az üzleti partnerek, a független fejlesztők és a vállalati IT fejlesztők Microsoft.NET szolgáltatásokat készíthessenek.

A fejlesztőknek a Microsoft a Microsoft.NET fejlesztőeszközök olyan új családját hozza létre, amelytől egészen az alaptól terveznek meg a web, az ügyfél- és szerveralkalmazásokat, illetve a szolgáltatások számát. Ezekkel az eszközökkel a fejlesztőknek lehetőségük nyílik a jelenlegi, az adatokat statikusan megjelenítő webet úgy átalakítani, hogy az interaktív szolgáltatások hangsúlyos szerepet kap-

janak. A Microsoft áttörést jelentő Visual Studio eszközkészlete automatizálta a webszolgáltatások készítését a gyors alkalmazáskészítést segítő drag-and-drop módszerrel, amelyben úttörő szerepet játszott a Visual Basic fejlesztőrendszer. Ezek a szolgáltatások már bármely XML-t értő platformon használhatók. A Visual Studio az XML kódot automatikusan készíti el. A Microsoft egy új BizTalk eszközkészlet megjelenését is hirdeti, amellyel, a megfelelő szolgáltatások összeállításával, vizuálisan programozhatunk üzleti folyamatokat. A vállalati elemzők így azokkal a módszerekkel hozhatnak létre megoldásokat, amelyekkel a fejlesztők is dolgoznak.

A Microsoft.NET programozási modellben a független fejlesztőknek kevesebb erőforrásra kell koncentrálniuk annak függvényében, hogy az alkalmazások hol és hogyan futnak, de leginkább, hogy mit csinálnak, vagyis, hogyan tudnak valódi értéketvélni. A Microsoft.NET a fejlesztőket érintő nagy kihívások közül például a kezelhetőség és a használhatóság közötti eltolódással is foglalkozik. Az alkalmazásszolgáltatás új elemekkel bővül: bármely más alkalmazással integrálhatók, teste szabhatók, programozhatók és offline módban is futtathatók.

Mivel a legfontosabb Microsoft.NET szolgáltatások előfizethetők, a fejlesztők dönthetnek a vásárlás vagy a saját megoldás mellett, annak függvényében, hogy mire akarják fordítani a fejlesztési erőforrásokat. Egyesek szavazhatnak a házi fejlesztésre, de a legtöbben valószínűleg egy jól elkészített megoldáscsomag mellett teszik le a voksot, amelyhez komoly fejlesztőeszköztámogatás is tartozik. Kevesen írnak a Windowsos saját nyomtatómeghajtót vagy ablakkezelő rendszert. A legtöbben ehelyett inkább a magasabb szintű saját termékeik differenciálására koncentrálnak.

A legfontosabb Microsoft.NET szolgáltatásmódulok kínálatát a következők lesz:

- ☞ **Azonosítás:** A Microsoft Passport és Windows hitelesítési technikára épülv a hitelesítés többféle szintjét nyújtja, a jelszavaktól és a tárcáktól kezdve az elektronikus hitelesítő kártyákig és a biometrikus eszközökig. Segít olyan szolgáltatásokat készíteni, amelyek biztosítják a testreszabhatóságot és az adatbiztonságot az ügyfeleknek, akik így biztonságosabban férhetnek hozzá a szolgáltatásaikhoz, függetlenül attól, hogy hol vannak, vagy milyen eszközt használnak.
- ☞ **Értesítés és üzenetküldés:** Egyesíti az azonnali üzenetküldést, az elektronikus levelezést, a faxot, a hangpostát és az értesítés, illetve üzenetküldés más formáit egy egységes megjelenésű, PC-re vagy más kommunikációs eszközre alkalmazható megoldással. A webalapú Hotmail elektronikus levelezési szolgáltatásra, az Exchange-re és az Instant Messenger-re épül.
- ☞ **Testreszabás:** Lehetőséget ad a felhasználóknak szabályok és preferenciák készítésére, amelyek implicit és explicit módon meghatározzák, hogyan kell kezelni az értesítéseket és üzeneteket, az adatmegosztási kérélmeket, és az olyan helyzeteket, amikor több eszközt is használnak (például a laptop mindig legyen szinkronizálva a Microsoft.NET tárolószolgáltatással). Az adatok

feltöltése egy új PC-re így egyszerű és gyors lesz.

- ☞ **XML tároló:** Univerzális nyelvet (XML) és protokollt (SOAP) használ az adatok jelentésének leírására. Így lehetővé válik az adatok integritásának megőrzése akkor is, ha azok átvele és kezelése több webhely és felhasználó között történik. Ennek eredményeképpen a webhelyek rugalmas, interakcióra képes szolgáltatásokká válnak, amelyek képesek egymás adatainak felhasználására. A Microsoft.NET az adatok tárolására egy biztonságos és címmezhető helyet is kínál. Ez bármely eszközzel elérhető, miközben replikációval a hatékonyság és az offline használat is biztosítható. Mások tárolóhelyének használata a tulajdonos beleegyezésével lehetséges. Egyesíti az NTFS, az SQL Server, az Exchange és az MSN szolgáltatásokat.
  - ☞ **Naptár:** A felhasználói vezérlés sarkalatos pontja az idő. Mikor lehet valakit zavarni, és mikor hagyják őt békén? Ez a kérdés egyre fontosabb lesz, ahogy egyre több eszközt egyre több ideig használunk, és ahogy a felhasználók és a szolgáltatások közötti kapcsolat egyre gyakoribbá válik. A Microsoft.NET biztosítja az általános, munkahelyi és otthoni naptárak egyesítését, úgy, hogy azok bármely eszközzel, más szolgáltatások, egyének számára is elérhető legyenek. Az üzenetküldésre és együttműködésre alkalmas Outlook ügyfélprogramja és a Hotmail naptárra épül.
  - ☞ **Címtár és keresés:** A Microsoft.NET lehetővé teszi azoknak a személyeknek és szolgáltatásoknak a megtalálását, amelyekkel fel kell venni kapcsolatot. A Microsoft.NET címtárak többek egyszerű keresőmotoroknál vagy „arany oldalaknál”. Programozható módon képesek kapcsolatba lépni szolgáltatásokkal, és választ tudnak adni egyes számaalapú kérdésekre is, amelyek a szolgáltatások képességeire vonatkoznak. Más szolgáltatásokkal együtt is alkalmazható, illetve összevonható vagy testreszabható.
  - ☞ **Dinamikus kézbesítés:** Biztosítja a Microsoft és a fejlesztők számára, hogy a funkcionálisitá inkrementálisan növeljék, és igény szerinti megbízható, automatikus frissítést hajtsanak végre anélkül, hogy a felhasználónak bármit is telepítenie vagy beállítania kellene. A Microsoft.NET bármely eszközön proaktív módon alkalmazkodik ahhoz, amit a felhasználó tenni szándékozik. Egy olyan világban, ahol a felhasználók több eszközön szeretnék élvezni a szolgáltatások előnyeit, a hagyományos, telepítésfüggő alkalmazásmo-dell fordítottjára van szükség.
- A Microsoft.NET elosztott szolgáltatásai online és offline is elérhetőek lesznek. Egy helyi kiszolgáló által biztosított, vagy az Interneten elérhető szolgáltatás online önálló gép-en is használható lesz, amelyik nem kapcsolódik az Internetre. A különféle elemek együttműködhetnek, vagy adatokat cserélhetnek egy „szövetségnek” nevezett folyamattal segítségével, amely a szervezetek számára lehetővé teszi, hogy eldöntés, saját, vagy egy külső, más által szolgáltatott infrastruktúrát használjanak anélkül, hogy meg kellene alku-dniuk a szolgáltatások szabályozása vagy



hozzáférése tekintetében, akár az Interneten, akár más-  
képpen veszik igénybe. Így például egy vállalati cím-  
szolgáltatás „szövetségre léphet” más szolgáltatással az  
Interneten. A Microsoft.NET-en alapuló szolgáltatások  
messze túlhaladjak azt, amit a mai internetalapú szol-  
galtatások kínálni tudnak.

A Microsoft.NET szolgáltatásmóduljai minden XML-t támogató  
platformon használhatók. A Windows biztosítja a legmegfele-  
lebb környezetet a webszolgáltatások készítésére és kézbesíté-  
sére, míg a Windows-alapú ügyfélprogramokat úgy optimali-  
zálják, hogy a webszolgáltatások mindenféle eszközön hasz-  
nálhatók legyenek. A Microsoft Windows DNA 2000 rendelke-  
zett először átfogó, XML-t támogató infrastruktúrával, amely-  
vel webszolgáltatások készíthetők és üzemeltethetők.

### Microsoft.NET felhasználói felület: intelligens kap- csolatkezelés

A számítástechnika jelenleg két világra osztható: a PC-ken és  
más eszközökön futó alkalmazások világára és a webhelyek  
világára. A Microsoft.NET lehetővé teszi a két világ közti  
együttműködést, miközben párosítja a gazdag funkcionalitást  
az Internet végtelen adattengerével. A mai webet a Tim  
Berners-Lee által előrevetített „interaktív térére” alakítja át.  
Az online és offline környezetben való munka ma kellemet-  
len és nem túl hatékony – még akkor is, ha csak egyetlen  
PC-vel dolgozunk. Egyáltalán nem integrált a web-  
böngészés *(csak olvasás)*, az alkotás *(dokumentumok írása  
és szerkesztése)*, a kommunikáció *(elektronikus levél, azo-  
nali üzenetküldés)*, a kalendárium és címjegyzék *(offline,  
eszközfüggetlő)*. Mind más alkalmazást igényel, amelyek  
funkcionalitása és kompatibilitása erősen változó. Legtöb-  
bünk egyetlen olyan egyesített, eszközfüggetlen környezet-  
szeretet, amely ahhoz a környezethez alkalmazkodik,  
amelyben éppen dolgozunk, transzparensen változtatva a  
helyi és távoli szolgáltatások, illetve alkalmazások között  
– ez lenne az internetkor univerzális felülete. Ennek meg-  
valósításához a Microsoft.NET a következőket kínálja a fel-  
használóknak:

☞ **Természetes kezelőfelület:** Olyan technológiák gyű-  
teménye, amelyek az ember és számítógép közti kom-  
munikáció új generációját valósítják meg – ide tartozik  
a beszéd, a látvány, a kézírás, és egy újfajta adatbeviteli  
ablakban, természetes nyelven történő bevitel. Ezek a  
megoldások kombinálhatók egy felhasználói felületben.  
A természetes kezelőfelület bármely eszközön  
vagy környezetben biztosítja a megfelelő felhasználói  
felületet.

☞ **Univerzális felület (canvas):** XML-el ötvözött infor-  
mációs architektúra, amely egyesíti a böngészést, a  
kommunikációt és a dokumentumok készítését egyet-  
len egységesített környezetben, egységes módon téve  
lehetővé az adatok elérését és szintetizálását. Az uni-  
verzális felület az XML-re épülő átalakítja a csak ol-  
vasható Internetet írható-olvashatóvá, ami a felhasz-  
nálók számára megnyitja az utat az adatok interaktív  
böngészése, szerkesztése, jegyzetekkel való ellátása és

vizsgálata előtt. Mivel az XML szolgál alapul, az univer-  
zális felület a különböző forrásból származó adatokat –  
a világ bármely részéről is származóknak – képes egye-  
síteni, megvalósítva ezzel az adatok transzparens hoz-  
záférést, szintetizálását és használatát.

☞ **Adatgyűnyök:** Kezeli a felhasználó személyazonosságát  
az Interneten, és fokozottabb ellenőrzési lehetőséget  
biztosít a webhelyek és -szolgáltatások felhasználóival  
való kapcsolatkezelése felett. Kezeli a meglátogatott  
oldalak listáját, a kontextust és a preferenciákat – a  
felhasználó múltját, jelenét és jövőjét az Interneten.  
Támogatja az adatbiztonsági technológiákat, például a  
PgP-t. A mai Internettel ellentétben az adatok a  
felhasználó ellenőrzése alatt maradnak, és ő dönt arról,  
ki férhet hozzá, és ki nem. Elég az egynégy preferen-  
ciákat egyszer elkészíteni, majd engedélyezni hasz-  
nálatát a kiszemelt webhelyek vagy szolgáltatások számára.  
☞ **Smarttag:** Az IntellSense technológia kiterjesztése  
a webtartalomra, ami lehetővé teszi, hogy a PC és más  
eszközök intelligensen kezeljék az Internetről vett  
adatokat. A kiterjeszhető architektúra bárki számára  
biztosítja az alkalmazkodó felhasználói felület és  
adatkezelők elkészítését. Az XML sémák alapja.

Az intelligens eszközök új nemzedékével dolgozva a  
Microsoft.NET-et a felhasználó ott használja, ahol akarja.  
Az internetes eszközök új generációját úgy tervezik, hogy  
képesek legyenek mások szolgáltatásait használni, és jó  
adatfeldolgozó-kapacitással rendelkezzenek. A hálózatot  
intelligensen használják, kiaknazzák a széles sávú kapcsola-  
tokat, de takarékosak a vezeték nélküli kapcsolatoknál. A  
programozhatóság, a testreszabás és az automatikus frissíté-  
s lehetősége, illetve az adminisztrációs igény eltűnése  
miatt ezek az új, intelligens eszközök robbanásszerűen el-  
terjednek a következő öt évben, és méltó társai lesznek a  
legjobb intelligens, internetes eszköznek: a PC-nek.

### Microsoft.NET: a termékek és szolgáltatások új generációja

Hosszú távon az összes alkalmazás szolgáltatásként lesz el-  
érhető, amelyre az Interneten lehet majd előfizetni. A Micro-  
soft és a többi alkalmazásszolgáltató számára ez annyit  
jelent, hogy színvonalasabban tudják a fogyasztókat kiszol-  
gálni, a telepítés és a biztonsági mentés központosított  
lesz, és pozitív visszacsatolás jön létre a termékfejlesztés-  
hez. A szolgáltatásként elérhető alkalmazások azt is lehe-  
tővé teszik a Microsoft és más független fejlesztők száma-  
ra, hogy gyorsabban tudják megoldani a biztonsági mentés  
és a vírusvédelem problémáit.

Úgy képzeljük, hogy alkalmazási szoftvereink többsége idő-  
előfizethető szolgáltatásá alakul, miközben továbbra  
is kínáljuk meglévő platformjainkat és alkalmazásainkat. A  
Microsoft azonban már kezdtető fogva ajánlani fog egy  
sor .NET terméket. Ezek közül néhány:

☞ **Windows.NET:** A Windows új generációja, amely előse-  
gíti a termelékenységét, az alkotást, a kezelhetőséget,  
a szórakoztatást és még sok más. Úgy tervezték, hogy  
a felhasználók teljes mértékben ellenőrzésük alatt tart

hasák életük digitális részét. A .NET legfontosabb szolgáltatásmoduljaival szorosan integrálták, így amellet, hogy személyre szabható, beépített támogatást nyújt a digitális anyagok és az együttműködés számára. Programozhatók a .NET szolgáltatásokkal, mint például az MSN.NET, a bCentral for .NET és az Office.NET. A Windows.NET sokoldalú platformot biztosít a .NET alkalmazások és szolgáltatásokat író fejlesztők számára. A Microsoft továbbra is kínálja és támogatja a Windows platform különféle verzióit .NET szolgáltatások nélkül.

🔗 **MSN.NET:** Ötvözi az MSN legfontosabb tartalmi elemét és szolgáltatásait a .NET platformmal. A felhasználóknak így lehetősége nyílik egyedi digitális személyazonosítók készítésére, illetve az intelligens szolgáltatások elérésére, amellyel biztosítható az adatok, szórakozási lehetőségek és személyek konzisztens, transzparens és biztonságos elérése, bármikor, bárhol és bármely eszközről. Az MSN.NET egy jelenleg béta fázisban lévő ügyfélprogramra fog épülni.

🔗 **Egyéni előfizetési szolgáltatások:** Az MSN.NET mellett a Microsoft olyan .NET-alapú, fogyasztóközpontú szolgáltatásokat is készít, amelyek a jelenlegi szórakoztató- és játékprogramokra, illetve oktatásban és termelésben résztvevő termékekre épülnek. A felhasználók számára ezek a szolgáltatások a hagyományos munkaszabványok alkalmazások előnyeit a felhasználói felületet új .NET családjának rugalmasságával, integrációjával és helyváltoztatási támogatásával képesek ötvözni.

🔗 **Office.NET:** Fejlett kommunikációt és termelést elősegítő eszközök. Ilyen például az univerzális felület, amely a kommunikációt, a böngészést és a dokumentumszerkesztést egyetlen környezetbe olvasztja, és ezzel biztosítja, hogy a felhasználók egységes módon tudják elérni az adatokat. Az univerzális együttműködési szolgáltatásokkal bárki együttműködhet más vállalati dolgozókkal, vagy akár vállalaton kívüli személyekkel is. Intelligens ügyfélprogramokra és szolgáltatásokra épülő, új architektúra, amely sokféle képességgel, nagy teljesítménnyel és bármely eszközön automatikus telepítési lehetőséggel rendelkezik. A Microsoft továbbra is kínálja és támogatja a .NET szolgáltatások nélküli Office verziókat.

🔗 **Visual Studio.NET:** Az MSDN és a Windows DNA kiszolgálók által teljes mértékben támogatott, XML alapú programozási modell és eszközök. Megkönnyíti a nagy mértékben elosztott, programozható szolgáltatások készítését, amelyek magukban álló gépeken, vállalati adatközpontokban és az Interneten futnak.

🔗 **bCentral for .NET:** Élvoalbeli, előfizetésalapú szolgáltatások és eszközök kis- és növekvő vállalatoknak. Magába foglalja a külső szolgáltatásként működő üzenetküldést és elektronikus levelezést, fejlett kereskedelmi szolgáltatásokat és a vevőkapcsolatokat kezelő új szolgáltatást, amely a .NET platformra épül. A kisvállalatok ezek segítségével színvonalasabb online szolgáltatást

nyújthatnak vásárlóiknak. Támogatja a külső katalógusok használatát, és figyelemmel kíséri a fogyasztók tevékenységét a személyre szabott szolgáltatás megvalósítása érdekében.

### Következtetés: a .NET forradalom

Tíz évvel ezelőtt a Microsoft olyan világot álmotott meg, amelyben az adatok szinte „megérinthező közelségbe” kerülnek hozzánk. Akkoriban az adatok elérése 4800 baud sebességű kapcsolaton folyt, az üzeneteket nem elektronikus levélben, hanem faxon küldték, és még csak kevesen hallottak az Internetről. Bár ebben a világban a felhasználók határozzák meg, hogy milyen adatokat, honnan és milyen eszközről akarnak elérni, akkor még nem lehetett tudni, hogy ennek megvalósításában milyen eszközök segítenek majd bennünket. Ma már tudjuk.

A Microsoft.NET platform a XXI. század első évtizedében forradalmasítja a számítástechnikát és a kommunikációt, mivel elsőként használja ki teljes mértékben mindkettő előnyeit. A számítástechnika és a kommunikáció egyszerűbbek lesznek, mint valaha. Internetszolgáltatások új generációját kelti életre, és sok ezer fejlesztő hozhat létre forradalmian új, online szolgáltatásokat és üzleteket. A felhasználóknak több irányítási lehetőséget ad a kezébe, amellyel jobban ellenőrizhetik a digitális azonosítójukat és a magánjellegű adataikat. Ami ezt lehetővé teszi: a szoftver.

A Microsoft.NET akkor lesz sikeres, ha sikerében mások is jelentős szerepet vállalnak. A Microsoft üzleti filozófiája mindig az olcsó és jó teljesítményű szoftverek tömeggyártása volt, amik segítik a magán- és üzleti felhasználókat tevékenységükben, továbbá vásárlóink, partnereink és a független fejlesztők számára új lehetőségeket tárnak fel. Ez a filozófia különbözteti meg a Microsoftot versenytársaitól, amit a Microsoft.NET tovább erősít.



Microsoft  
.net





### A DNS (Domain Name System – tartománynév-szolgáltatás) használata

A tartománynév-szolgáltatás vagy más néven DNS használata minden Internethez csatlakozó szervezet életében létfontosságú. A DNS hozzárendelést végez a köznapi nevek (például [www.netacademia.net](http://www.netacademia.net)) és a kommunikációhoz használt IP-címek között. Az Active Directory a DNS szolgáltatásain alapul: a DNS segítségével keressük meg például az Active Directoryban szereplő számítógépek IP címét, s ezzel magát a gépet. Ez alapvető változás a korábbi Windows operációs rendszerekhez képest, melyek NetBIOS neveket rendelnek az IP-címekhez, s amelyek WINS, vagy egyéb NetBIOS névhozzárendelési szolgáltatásra támaszkodnak.

Az Active Directory a Windows 2000 alapú DNS kiszolgálókkal működik együtt a leghatékonyabban. A Microsoft vezérlők segítségével könnyíti meg az áttérést a Windows 2000 alapú DNS kiszolgálókra, ezek végigvezetnek a teljes folyamaton. Más DNS kiszolgálók is használhatók, de ebben az esetben több időt kell majd töltenünk a DNS adatbázisok kezelésével. Ha úgy döntünk, hogy nem Windows 2000 alapú DNS kiszolgálókat használunk, gondoskodnunk kell arról, hogy lehetőleg támogassák az új dinamikus DNS frissítés (DNS dynamic update vagy DDNS) protokollt. Az Active Directory kiszolgálók ezzel frissítik mutatórekordjait, a munkaállomások pedig ezen rekordok segítségével keresik meg a tartományvezérlőket. Ha a dinamikus frissítés nem támogatott, akkor az adatbázist kézzel kell frissíteni. A DDNS protokollt a 2136-os RFC írja le.

A Windows tartományok és az internetes tartományok mostantól teljesen fedik egymást. A tartománynevek (mint például a [vallalat.hu](http://vallalat.hu)) olyan Active Directory tartományvezérlőket azonosítanak, melyek az adott tartományért felelősek, így bármely DNS eléréssel rendelkező munkaállomás képes megtalálni egy adott tartományvezérlőt. Az Active Directory ügyfelek a DNS hozzárendelés segítségével bármely szolgáltatást képesek megtalálni, mert az Active Directory kiszolgálók - dinamikus frissítés használatával - bejegyzik önmagukat a DNS-be. Ezek a címek mind a tartományt, mind a nyújtott szolgáltatást azonosítják, és szolgáltatásforrás-rekordokon (Service Resource Record – SRV RR) keresztül kerülnek közzétételre. Ha SRV RR-ek az alábbi formátumot követik:

### szolgáltatás.protokoll.tartomány

Az Active Directory kiszolgálók LDAP protokollon keresztül teszik elérhetővé a címtárat, ezért ha egy kiszolgálót keres a [vallalat.hu](http://vallalat.hu) tartományban, akkor a DNS-rekordban az [ldap.tcp.vallalat.hu](http://ldap.tcp.vallalat.hu) rekordot fogja keresni.

**Az Active Directoryval kapcsolatos problémák 110%-a a DNS-re, illetve a DNS ismeretének hiányára vezethető vissza!**

### Replikáció

Azok a rendszergazdák, akik Active Directory-t építenek, hamar észreveszik majd, hogy a hálózat jelentős mértékben támaszkodik az Active Directory szolgáltatásaira. Ez azt jelenti, hogy az Active Directorynak több kiszolgálón is elérhetőnek kell lennie, így ha valamelyik meghibásodik, a munkaállomások kapcsolatba tudnak lépni egy másik kiszolgálóval, melyben rendelkezésre áll a szolgáltatások és az adatok másolata. A Windows NT korábbi verzióival használt Windows NT tartományadatbázisoktól eltérően az információk replikálása, melynek Active Directory kiszolgálóknak elküldhetők. Bár ez a replikáció folyamatát bonyolultabbá teszi, annak az esélyét is kiküszöböli, hogy egy tartományvezérlő meghibásodása az adatbázist módosíthatatlanná tegye. Az új módszer a tartományvezérlők terhelését is csökkenti. A replikáció folyamatának megkezdéséhez elegendő egyszerűen hozzáadni a tartományvezérlőket egy Active Directory tartományhoz.

A redundáns kiszolgálók megfelelő működésében az egyik legbonyolultabb feladat az információk replikálása, illetve annak biztosítása, hogy minden kiszolgáló a legfrissebb tartalommal rendelkezzen. Az Active Directory többforrású (multimaster) replikációt alkalmaz, amely garantálja, hogy a frissítések minden Active Directory kiszolgálón megjelenjenek. Minden kiszolgáló számon tartja, hogy melyik frissítést melyik kiszolgálótól kapta, és meghibásodás esetén intelligens módon csak azokat a frissítéseket kéri újra, amelyekre tényleg szüksége van.

### Az Active Directory replikáció működése

Az Active Directory replikációja azok számára fog igazán logikusnak tűnni, akik már ismerik a replikáció működését a Windows NT 4.0 tartományokban. Minden frissítés kap egy 64 bites egyedi sorozatszámot (USN – Unique Sequence Number), melynek értéke minden módosítással alkalmával növekszik. Ezek a számlálók kiszolgálóspecifikusak, azaz minden Active Directory kiszolgáló saját számlálóval rendelkezik.

Amikor egy kiszolgáló más Active Directory kiszolgálóra replikál egy frissítést, a módosítással együtt elküldi az USN-t is. Minden kiszolgáló rendelkezik egy belső listával, mely a replikációs partnereket, illetve az azoktól kapott magasabb USN értékeket tartja nyilván. A frissítést kapó kiszolgáló csak azokat a módosításokat kéri el, amelyeknek USN értéke magasabb a korábban kapott értékeknél. Ennek a módszernek többek közt az az előnye, hogy meggátolja a frissítések vég nélküli keringését az Active Directory kiszolgálók között.

A multimaster replikációs sémában rejlik egyik probléma, hogy egyazon objektum frissítése egyszerre több helyen is megtörténhet. Ha például egy rendszergazda Budapesten az egyik felhasználónevet „Egerfalvi”-ről „Egerfalvi-ra” változtatja, és egy másik rendszergazda ugyanekkor Pécsen ugyanennek a felhasználónak a nevét „Egerfalvi”-ről „Egerfalvi-ra”, akkor replikációs ütközés lép fel, melyet automatikusan észlelni és javítani kell.

Az Active Directory ún. tulajdonságverzió-számokat tárol, melyek lehetővé teszik a replikációs ütközések felderítését. Ezek a számok az Active Directoryban szereplő minden ob-

jektum minden tulajdonságára nézve különbözőek, és minden alkalommal frissülnek, amikor a tulajdonság módosul. Ezek a számok a változással együtt terjednek az Active Directory-n keresztül, így ha egy kiszolgálóhoz egyazon tulajdonság két különböző frissítése érkezik ugyanazzal a tulajdonságverzió számmal, akkor a kiszolgáló képes megállapítani, hogy replikációs ütközés történt.

Az Active Directory kiszolgálók úgy oldják fel az ütközést, hogy a későbbi időbélyegzővel rendelkező frissítést alkalmazzzák. Az időbélyegzőt a módosítást kezdeményező kiszolgáló hozza létre, ezért nagyon fontos, hogy a rendszeridő szinkronizálva legyen a Windows 2000 kiszolgálók között. A kiszolgálók óráinak összehangolása a beépített SNMP időszinkronizáló szolgáltatás segítségével automatikusan megvalósul!

### Partícionálás

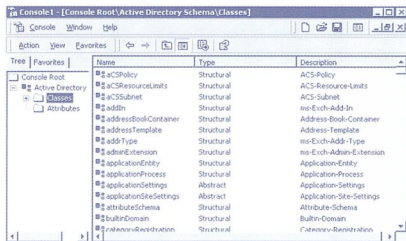
A nagy hálózatok több százezer objektumot tartalmazhatnak. A Windows NT esetében ilyen mennyiségű objektum kezeléséhez több tartományra volt szükség. A felhasználókat és az erőforrásokat különálló tartományokra osztottuk, a tartományok közt pedig megbízott kapcsolatokat hoztunk létre. Az adatbázisok szerkezete egyszerűen nem tette lehetővé, hogy az objektumok száma több százezerre növekedjen. Az Active Directory tartományokban a méretkorlátok szerencsére sokkal kevésbé számítanak. A nagyon nagy méretű Active Directory támogatása azonban egyetlen tartományvezérlőre hihetetlen mértékű terhet róhat. (Lásd *előző számunkban a nagyméretű címűtár építéséről szóló cikket!* – a szerk.)

Ezen terhelés csökkentése érdekében az Active Directory partícionálható. A partícionálás lehetővé teszi, hogy a különböző tartományvezérlők az adatbázis különböző részeit kezeljék, így csökkentve az egyes kiszolgálókra jutó terhelést. A munkaállomások továbbra is egységesen használhatják a különböző Active Directory partíciókban elhelyezkedő erőforrásokat, így a rendszergazdák anélkül képesek kezelni a nagy méretű Active Directory tartományokat, hogy a tartományvezérlőknek az egész adatbázist kezelniük kellene.

### Séma: attribútumok és objektumosztályok

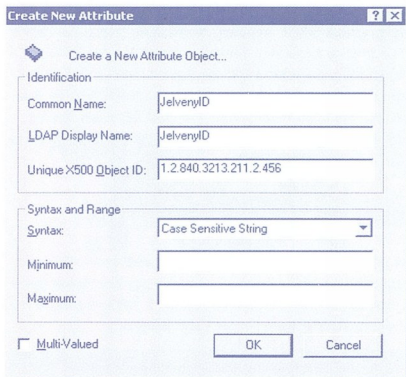
Ahogy az korábban már láttuk, a séma olyan attribútumok halmaza, amelyek egy adott objektumosztályt írnak le az Active Directoryban. A különböző objektumosztályok esetében különböző adatok nyomon követésére van szükség, ezért olyan nagy a sémák jelentősége. A „felhasználók” objektumosztály leírásához például a következő attribútumok szükségesek: vezetéknev, utónév, telefonszám, elektronikus levél-cím és hagyományos levél-cím. A „nyomatok” objektumosztály leírásához más attribútumok szükségesek: azt szeretnénk tudni, hogy az adott nyomtató mennyire gyors, vagy képes-e kétoldalasan, illetve színesben nyomtatni. Ezek az attribútumok az Active Directory Schema MMC beépülő modul segítségével tekinthetők meg és szerkeszthetők. Az Active Directory Schema nem rendelkezik ikonnal a Start menüben. Tulajdonképpen eléggé veszélyes dolog meggondolatlanul babrálni a sémát, mert nemcsak hogy az egész erődben elterjednek a módosítások, hanem ráadásul visszavonhatatlanok is. Talán emiatt tüntették el a Start menüből, sőt, még a beépülő mo-

dul sincs regisztrálva – pedig ott csücsül a WINNT\SYSTEM32 könyvtárban. Indításához először regisztrálni kell az MMC modult (*parancssorból adjuk ki a REGSVR32 SCHMMGMT.DLL parancsot*), majd el kell indítani az MMC felületet, és hozzá kell adni az Active Directory Schema nevű modult.



### Az Active Directory Schema segítségével módosíthatók az osztályok és az attribútumok

Most pedig evezünk veszélyesebb vizekre. Senkinek sem ajánlom azonban, hogy az éles, üzembe állított AD kiszolgálóján hajtsa végre a sémamódosítást, mert mint mondtam, a dolog végleges, visszavonhatatlan. Az objektumosztályokhoz alapértelmezés szerint egy olyan logikai attribútumhalmaztartozik, amely a legtöbb szerkesztési igényeinek megfelelő. Azonban sok szervezetnek szükség lesz arra, hogy bizonyos objektumokkal kapcsolatban további információkat kövessen nyomon. Ha például az alkalmazottak jelenviszámmal rendelkeznek, érdemes lehet (*ki tudja?*) ezt az információt is felvenni az objektumosztályba. Első lépésként létre kell hozni egy JelvényID nevű attribútumot, amint az az alábbi ábrán látható. Második lépésként az új attribútumot opcionálissá kell tenni a „felhasználók” osztály számára.



### Az Active Directory Schema beépülő modul segítségével új attribútumok vehetők fel.



A sémát az Active Directory úgy tárolja, mint bármely más objektumot. Így a séma örökli a tartományon keresztül történő automatikus replikálódás képességét, kihasználja az Active Directory biztonsági szolgáltatásait is, és lehetővé teszi, hogy séma fölötti jogköröket delegáljunk a különböző felhasználóknak és csoportoknak. A sémaobjektumok ACL-einek (*hozzáférés-szabályozási listáinak*) módosítása révén a rendszergazdák bármely felhasználó számára lehetővé tehetik egy objektumosztály attribútumainak megváltoztatását vagy bővítését.

Az új attribútumok számos tulajdonsággal rendelkeznek, melyeket be kell állítani. Meg kell adnunk az attribútum nevét (például *JelvényID*), a tárolandó adat típusát (például *karaktérlánc* vagy *szám*), illetve az érték tartományt (például *a karaktérlánc hosszúságát*). Egyedi objektumazonosítót (*OID – Object Identifier*) is meg kell adni (lásd később). Az új attribútumok indexelhetők, ekkor bekerülnek a globális katalógusba. Olyan attribútumokat érdemes indexelni, amelyekben a felhasználók keresni fognak. Az említett példánál maradva, ha a biztonsági szolgálat embereinek szüksége van arra, hogy a felhasználói fiókokat a jelvényazonosító számok alapján keressék meg, akkor ezt az attribútumot indexelni kell. Az index-szel nem rendelkező attribútumokban történő keresés lassú és processzorigényes feladat, melynek során az egész címtárfát be kell járni.

Az osztályok és az attribútumok sem az Active Directory Schema sem más eszköz segítségével nem törölhetők. Létrehozásuk után örökké létezni fognak az Active Directoryban. Az egyetlen lehetőség az osztály deaktiválása, amely megakadályozza annak további használatát. Nem deaktiválható olyan osztály vagy attribútum, amely függőséggel rendelkezik az Active Directory-ban. Ha például az egy attribútumot még használ egy aktív osztály, akkor annak az attribútumnak aktívának kell maradnia.

### Honnan származnak az objektumtípus azonosítók?

Az objektumtípus azonosítók nem a GUID-k (*amelyeket az Active Directory is képezni tud*), hanem az X.500 szabvány-nak megfelelő, iparági besorolászerű azonosítók (*OID*), melyekkel elkerülhető, hogy két cég azonos bővítményeket készítsen az Active Directoryhoz, melyek azután nem lennének telepíthetők közös erődbe. Az objektumtípus azonosítók globális egyedisége csak egy központi ügynökség segítségével biztosítható, mely az azonosítók hozzárendeléséért felelős. Az Interneten ez már elterjedt gyakorlat: a tartományneveket az InterNIC, az IP-hálózatiakat pedig az Internet Assigned Numbers Authority (*IANA*) osztja ki. Objektumazonosítókat a National Registration Authority-tól (*NRA*) kérhetünk. Az NRA minden országban más és más, az Amerikai Egyesült Államokban az NRA szolgáltatást az Amerikai Nemzeti Szabványügyi Intézet (*American National Standards Institute – ANSI*) látja el. Az ANSI csekély térítés ellenében bármely szervezet számára biztosít egy gyöker-OID-t. A szervezeten belül létrehozott valamennyi objektum megkapja ezt a gyöker-OID-t mint előtagot, így az objektumazonosítók globális egyedisége garantált.

Az NRA-k listája a Nemzetközi Szabványügyi Szervezet

webhelyén a következő címen tekinthető meg: <http://www.iso.ch>. Ugyanakkor ahhoz, hogy játszadozzunk a sémával, valószínűleg túl sok lenne egy fillér kifizetése is. OID nélkül viszont nem vehetünk fel újabb attribútumot. A kibúvó a Resource Kitben található *OIDGEN.EXE*, mellyel nagy valószínűséggel egyedi, sehol máshol nem létező OID-eket lehet generálni. Ezek tökéletesen jók belső hálózatunk bővítéséhez, de mint ahogy belső IP címekkel nem sokra megyünk a nyílt Interneten, ugyanígy hajítófát sem ér a random-OID akkor, ha eladásra készülő címtárteremben próbáljuk felhasználni.

Az Active Directory sémájának kiterjesztése igen hatékony lehetőség, bár a Microsoft által biztosított alapértelmezéseken felül a legtöbb rendszergazdának soha nem lesz szüksége más osztályok és attribútumok használatára.

### Objektumok

Az első pillanatban sokakat zavarba ejt az objektumosztályok, attribútumok és az objektumok kapcsolata. Az objektumok egy objektumosztály alapján jönnek létre. Az attribútumok egy-egy objektumosztályt írnak le. Egy objektum létrehozásakor az objektum örökli objektumosztályának valamennyi attribútumát. A dolog megértését a következő tény nehezíti: az objektumosztályok és az attribútumok maguk is objektumok az Active Directoryban. Ezt a tényt szerencsére a legtöbb felhasználói felület elrejt.

Egy objektum lehet egy konkrét dologra utaló hivatkozás, vagy maga a tényleges információ. Például a felhasználói fiókokkal kapcsolatos valamennyi információ az Active Directory-ban van tárolva. Ugyanakkor az Active Directory a lemezököttekkel kapcsolatban csupán egy hivatkozást tárol. Bár a hivatkozás önmagában nem hasznos információ, mégis felhasználható a kötetek megkeresésére a fájlkiszolgálón. Új objektumosztályok létrehozásakor alaposan át kell gondolni, hogy az objektum vajon egy külső dologra utaló hivatkozást fog-e tárolni, vagy az objektum attribútumai tartalmazni fogják-e az összes fontos információt. Bár az Active Directory rendkívül kényelmes eszköz, nem szabad nagy mennyiségű állandóan változó vagy ritkán használt információ tárolására használni.

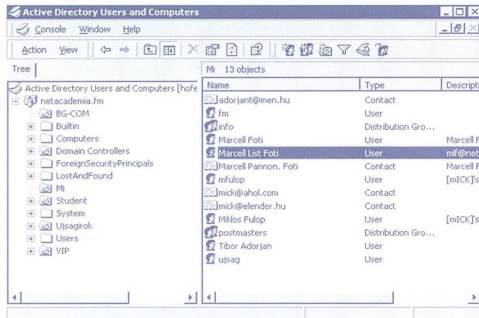
Amikor valaki új felhasználót vagy számítógépet ad az Active Directory-hoz, tulajdonképpen egy objektumot hoz létre. Az objektumok létrehozását gyakran *közvetéltelnek* is nevezzük, mert az objektum létrehozása replikációs folyamatot indít el, amelyben az új információ replikálódni kezd a tartományban található összes Active Directory kiszolgálóra.

### Szabványos objektumosztályok

A Windows 2000 Server az Active Directory segítségével tárolja a felhasználókra, csoportokra és számítógépfiókokra vonatkozó nagy mennyiségű adatot. Ezek az adatok különösen a rendszergazdák számára fontosak, mert ezek képezik az Active Directory leggyakrabban elért részét. A korábban Windows NT környezetben dolgozó rendszergazdák számára az új felhasználói felület esetleg nem tűnik túl intuitívnek, de rövid idő eltelté után a kezelése egészen egyszerűvé válik.

## Felhasználók

A felhasználói fiókok kezelése többé nem egy különálló eszköz segítségével történik, hanem az Active Directory Users and Computers MMC beépülő modul használjuk. Maguk a felhasználói fiókok is jelentősen megváltoztak. A Windows NT 4.0 csupán a felhasználói nevet, a teljes nevet, a leírást, a jelszót és néhány egyéb attribútumot tartott nyilván. A Windows 2000 Server rendszerben a címárknak köszönhetően ezek az attribútumok kiterjeszhetők. Az Active Directory segítségével számtalan személyes adat nyilvántartható, beleértve a telefonszámokat, a lakcímeiket és a felettesek neveit. Ezek a kiegészítő információk teljesen opcionálisak.

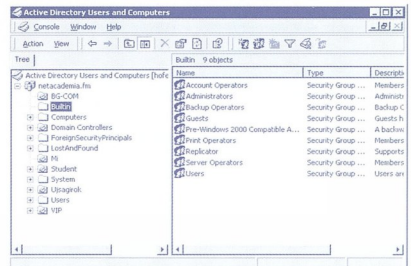


*Az Active Directory Users and Computers MMC modul felváltja a Felhasználókezelőt*

## Csoportok

Az Active Directory csoportok a Windows NT korábbi verzióiban megszokott felhasználói csoportokhoz hasonlóak, azonban számos új szolgáltatással rendelkeznek. A Microsoft Exchange legújabb verziója lehetővé teszi a csoportok elektronikus levelezési (terjesztési) listákként történő használatát. Hogy a szolgáltatás még hasznosabb legyen, a csoportokhoz elektronikus levelezési fiókok is hozzáadhatók, így a terjesztés azon felhasználók felé is lehetővé válik, akik nem tagjai ugyanannak az Active Directory fának. A csoportok egymásba is ágyazhatók, így a rendszergazdáknak lényegesen kevesebb időt kell tölteniük a felhasználók és a csoportok kezelésével. Mostantól három különféle csoport létezik. Az univerzális csoport a leg rugalmasabb típus, az erdő tetszőleges felhasználóját vagy csoportját tartalmazhatja. Ezek a csoportok a tartományon kívül is replikálódnak, és megjelennek a globális katalógusban. A globális csoportok csak saját tartományukból fogadhatnak be felhasználókat és csoportokat. A globális csoportok szerepelnek ugyan a globális katalógusban, de tagsági listájuk már nem. A tartomány szintű helyi csoportok csak egyazon tartományban található hozzáférés-szabályozási listákra (ACL) alkalmazhatók, de más tartományokba tartozó felhasználókat és csoportokat is tartalmazhatnak. Ezek nem replikálódnak a tartományon kívül, és nem jelennek meg a globális katalógusban. A három csoport típus mindegyike részt vehet a tartomány biztonsági be-

állításában, vagy működhet egyszerű terjesztési listaként. A Windows 2000 Server számos alapértelmezett csoportot tartalmaz, ezeket beépített csoportoknak nevezzük, és igény szerint további csoportokat vehetünk fel.



*A Windows 2000 Server számos beépített csoportot tartalmaz*

## Számítógépfiókok

A tartományhoz csatlakozó számítógépek automatikusan számítógépfiókot kapnak az Active Directoryban. Ez a folyamat ahhoz hasonlít, mint amikor egy új gépet adunk a Windows NT 4.0 tartományhoz. Azonban a gépeket akkor is a tartományhoz lehet adni, ha azok nem vesznek részt a tartományvédelemben. (Például létrehozhatunk egy számítógép-objektumot egy UNIX rendszer számára is.)

## Lightweight Directory Access Protocol (LDAP)

Az Active Directory tükrözi a Microsoft azon célkitűzését, hogy a rendszerek minél inkább támaszkodjanak a szabványos protokollokra. A Lightweight Directory Access Protocol (LDAP) az IETF (Internet Engineering Task Force) szabványa. A protokoll a munkaállomások és kiszolgálók címárakkal kapcsolatos információcserejét definiálja. A Windows 2000 Active Directory az LDAP 2-es és 3-as verzióját használja.

## Egyszerű felhasználónév

Az LDAP megkülönböztető nevek a számítógépek számára kiválóak, de a felhasználók nehezen jegyzik meg őket. Mivel már hozzászokunk az elektronikus levelezémeikhez, ezért az Active Directoryban ezeket a címeket a teljes objektumnévre mutató hivatkozásként használhatjuk. Az én bejelentkezési nevem például `marcellf@netacademia.net`.

A felhasználók egyszerű felhasználónevével segítségével jelentkezhettek be a Windows 2000 rendszerbe. Az emailszerző nevek felváltják a régebbi Windows hálózatokban használt felhasználóneveket, így a felhasználóknak nem kell teljes LDAP nevéket begépelniük. Az egyszerű felhasználónevek további előnye, hogy akkor sem változnak meg, ha a rendszergazda áthelyezi vagy átnevezi az adott felhasználói fiókot.



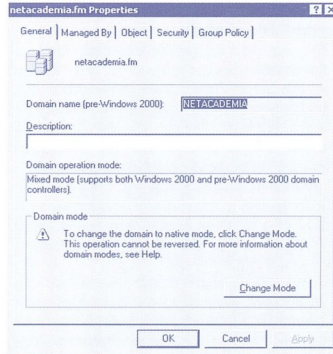
## ADSI (Active Directory Service Interface)

Az ADSI (Active Directory Service Interface) lehetővé teszi, hogy az alkalmazások anélkül lépjenek kölcsönhatásba bármely címtárszolgáltatással, hogy ismerniük kellene a megfelelő protokollok belső szerkezetét. A fejlesztők olyan programokat és parancsfájlokat írhatnak, amelyek az ADSI-t kihasználva írják és olvassák az örökölt Windows NT 4.0 címtárakat, NetWare NDS címtárakat, NetWare 3 binderyket és az LDAP címtárakat (pl. Active Directory). Akár anélkül is létrehozhatunk olyan alkalmazásokat, amelyek kihasználják a meglévő címtárakat, hogy előzetesen ismernénk a használt címtár típusát. (Erről a témáról pedig e számunk fejlesztői rovatában olvashatnak – a szerk.)

## Hálózattervezés az Active Directory-hoz

A munkaállomások a telephely-információk segítségével azonosítják a legközelebbi Active Directory kiszolgálót. Mivel a telephelyek IP-alhálózatoknak felelnek meg, minden alhálózaton el kell helyezni egy Active Directory kiszolgálót. Arról is gondoskodni kell, hogy az egyazon logikai alhálózaton található valamennyi rendszer a helyi hálózaton keresztül csatlakozzon. Bizonyos útválasztási technológiák – mint például a Proxy Address Resolution Protocol (Proxy-ARP) – lehetővé teszik, hogy az ugyanazon logikai alhálózaton található rendszerek a hálózat különböző fizikai szegmensein helyezkedjenek el. Ennek a beállításnak a hatására az ügyfelek egymáshoz közelebbinek érzékelik az egyes rendszereket, mint a valóságos helyzet, ezért érdemes a szabványos útválasztási technológiákhoz ragaszkodni. Ha az olvasónak mindez nem sokat mond, akkor minden esetben van: hálózata valószínűleg megfelelően van beállítva. A hálózat átállítása előtt meg kell tervezni az Active Directory szerkezetét. Erre két lehetőség nyílik: létrehozhatunk egy új fát, vagy csatlakozhatunk egy létező fához. Az átállítani kívánt hálózat első tartománya esetében nyilvánvalóan új fát kell létrehozni. Ha azonban a rendszergazda több tartományt szeretne egyesíteni egyetlen Active Directory tartományban, akkor a létező fa gyermekeként kell csatlakoznia.

Az átterítés mindig a Windows NT 3.51 vagy 4.0 PDC-vel (Primary Domain Controller – Elsődleges tartományvezérlő) kell kezdeni. Az aktuális tartomány felhasználói és csoportjai automatikusan átkerülnek az Active Directory-ba, a létező ügyfelek pedig pontosan úgy fognak kapcsolódni az új tartományvezérlőhöz, mintha az még mindig PDC lenne. Amíg az Active Directory kiszolgálók, és az örökölt BDC-k (tartalek tartományvezérlők) párhuzamosan működnek, a tartomány vegyes üzemmódban fog működni, amint az alábbi ábráról leolvasható. A vegyes tartományok nem tudják maradéktalanul kihasználni az Active Directory új szolgáltatásait, mert az Active Directorynak biztosítania kell a lefelé irányuló kompatibilitást. Vegyes üzemmódú tartományokban például nem használhatók egymásba ágyazott csoportok.



## Örökölt BDC-k létezőkor vegyes üzemmódú tartományok működnek.

Ha a rendszergazda meggyőződött arról, hogy a vegyes üzemmódú tartományok tökéletesen működnek, végre kell hajtania a BDC-k átállítását is. Miután minden tartományvezérlő átállítása megtörtént, a tartomány natív üzemmódba kapcsolható, és kihasználható az összes új szolgáltatás. A tagkiszolgálók és a munkaállomások támogatása teljes körű, nincs szükség semmilyen módosításra, vagy beavatkozásra az Active Directory kiszolgálókkal történő együttműködés biztosításához. A tagkiszolgálók frissítése további előnyökhöz jár, de először mindig a tartományvezérlőket kell frissíteni.

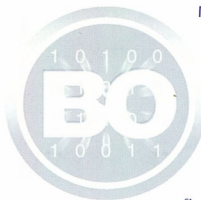
A Windows NT Workstation munkaállomások csak akkor tudják kihasználni az Active Directory új szolgáltatásait, ha azokat Windows 2000 Professional rendszerre frissítik. A Windows 95 és Windows 98 ügyfelekhez javítócsomag készült, melynek telepítése után ezek is képesek lesznek érzékelni az Active Directoryt, és részt tudnak venni a Kerberos adatvédelemben.

## Összefoglalás

A hálózatot leginkább az Active Directory bevezetése miatt érdemes frissíteni. Az Active Directory a Windows NT tartományokat egyesíti az internetes tartományokkal, és lehetővé teszi azok vállalati szintű kiterjesztését. Bár a legjelentősebb előny a csökkentett üzemeltetési költség, a felhasználók a globális katalógus továbbfejlesztett keresőlehetőségei révén közvetlen előnyökhöz is jutnak.

A Microsoft szándéka az, hogy az Active Directoryra történő átérés a lehető leegyszerűbb legyen. Varázslók állnak rendelkezésre, melyek segítségével a DNS-feladatok átadhatók a Microsoft DDNS (Microsoft DNS dynamic update) kiszolgálóknak. A felhasználók és a csoportok automatikusan importálódnak az örökölt Windows NT tartományokból. Végezetül az Active Directory telepítésének minden mozzanata intuitív, továbbá minden összetett feladat automatikus.





## Microsoft SQL Server 2000 Transact SQL – 1. rész Egyszerű lekérdezések

### Bevezetés

Az SQL Server 2000 a 2000. évi adatbázispiac leghangosabban debütáló terméke. A konkurencia is árgus szemekkel figyeli mit tud az új termék, és igyekszik, hogy elvegye tőle a leggyorsabb adatbáziszkiszolgálónak járó díjat. Nem véletlenül! Már az SQL Server 7 is nagyon kifinomult és hatékony relációs adatbáziskezelő volt, azonban az utód még igen sok és rendkívül érdekes szolgáltatással rukkolt elő. Ezek kiaknázásához azonban elengedhetetlen a Transact SQL ismerete.

A Transact SQL egy nagyon bonyolult dolog. Legalábbis legtöbbször ezzel nyugtatják magukat, amiért nem tanulják meg. Ebben a cikksorozatban rácáfoltuk erre az állításra. Nagyon kifinomult nyelv, azonban egyszerűbb lekérdezéseket és adatmódosító scripteket bárki képes lesz írni, amint elolvasta és végiggyakorolta ezt a cikksorozatot. Vágujunk hát bele!

### Előkészületek

Adott az SQL 2000 Server-ünk. Van rajta egy Northwind nevű adatbázis, ami a szerencsésnek a kiszolgálóval együtt települt fel. *(Aki kevésbé az, az futtassa le az `instnwnd.sql` scriptet a Program Files\Microsoft SQL Server\Install mappából. Ez létrehozza az adatbázist, és feltölti az adatokkal.)* Az adatbázis a Northwind Traders nevű ételkereskedelemmel foglalkozó cég ügyfeleit, eladásait és sok egyéb céges adatát rögzíti. Van benne egy Customers tábla, ami a cég ügyfeleit tartja nyilván, egy Products tábla, ami a kínált termékeket, egy Orders tábla, ami a megrendeléseket, egy Employees, ami az alkalmazottakat. Ezeken kívül van még sok egyéb tábla is benne, de az egyszerűség kedvéért most csak ezeket fogjuk használni.

Adjuk ki az első SQL parancsunkat! Nyissuk meg a Query Analyzer nevű programot a Start Menu/Microsoft SQL Server alatt. Ez egyike azon alkalmazásoknak, amelyet nagyon szeretünk használni. Ahogyan a Windows 2000 parancsokat a parancssorban próbálhatjuk ki, hasonló módon a Query Analyzer segítségével lehet tesztelni az SQL lekérdezéseinket.

Elíndult a program, és kérdezi, hogy melyik kiszolgálóra szeretnénk csatlakozni. Írjuk be a gépünk nevét. Ha a lokális gépen van, akkor írjuk *(local)*-t. Ha mi vagyunk a nagyfiúk a gépen *(administrator)*, akkor használjunk Windows NT authentication-t. Ha nem, akkor használjunk SQL authentication-t, sa login nevet és a megfelelő, leggyakrabban üres jelszót. *(Éles környezetben nem árt ám adni neki egy jelszót! Az esetek igen jelentős részében elfelelkeznek erről, és később csodálkoznak, hogy az SQL Servert is futtató webkiszolgálónkon reggelre kicserélték a főoldalt. Az sa login-nal az SQL Serverre belépve ugyanis kb. 30 mp. alatt rendszergazdává válhat bárki.)* Akik több SQL Servert is futtatnak ugyanazon a gépen, a gépnév/példány nevet írják be a név mezőbe.

### Bemelegítés

Sikerült csatlakozni? Akkor a hehezén már túl vagyunk. A Toolbar közepén van egy legördülő lista, ott válasszuk ki a Northwind adatbázist, jelezzük, hogy SQL parancsainkat abban szeretnénk végrehajtani. Adjunk a gépnek végre munkát! Kérjük meg, hogy szíveskedjen kilistázni a cég összes vásárlójának a nevét. Ez SQL-ül valahogy így néz ki:

```
SELECT
  CompanyName
FROM
  Customers
```

Írjuk be a parancsot a Query ablakba, majd nyomjunk F5-öt a futtatáshoz. Felszáll a mérőfűt! Ha minden rendben ment, akkor az alsó ablakban 91 céget láthatunk a világ minden részéről, élen a közismert Alfreds Futterkiste céggel:

```
Alfreds Futterkiste
Ana Trujillo Emparedados y helados
Antonio Moreno Taqueria
...
```

Hogyan fordíthatnánk le magyarra az előbbi SQL parancsot? Hát körülbelül úgy, mintha angolról fordítanánk. Válaszd ki a cégnevet a Customers táblából. Ez a szépsége az SQL nyelvnek, hogy nem For, Loop és egyéb csúfságokkal kell bibelődni, hanem mindjeden angolul megfogalmazzuk a kérésünket, és az SQL Server végrehajtja azt. A SELECT kulcsszó után megadjuk azoknak az oszlopoknak a nevét, amelyekre szeretnénk látni a parancs kimenetét, a FROM után pedig azoknak a táblának a nevét, ahonnan származnak az adatok. Ebben a példában csak egy forrás táblánk volt.

Néhány gondolat a parancsok tagolásáról. Mint minden programozási nyelvben, sokféle tagolási lehetőség kínálkozik a számunkra *(C programozók szoktak vére menően harcolni a kapcsos zárójelek helyéről)*. Az SQL parancsok akkor lesznek a legolvashatóbbak, ha minden kulcsszót külön sorba írunk, azaz a SELECT is kap egy saját sort, a WHERE is és a többiek is. A kulcsszavakhoz kapcsolódó egyéb paramétereiket is célszerű új sorba írni, valamelyest beljebb tolva. A cikk további részében ehhez a konvencióhoz tartom magam.

**Feltételek, és ahol a problémák kezdődnek:** a NULL fűszerezéjük meg a példánkat! Válogassuk le csak a német vásárlókat, azaz akiknek a Country mezőjében Germany áll. Mi sem egyszerűbb:

```
SELECT
  *
FROM
  Customers
WHERE
  Country = 'Germany'
```

| ALFKI Alfreds Futterkiste      | Maria Anders        |
|--------------------------------|---------------------|
| Sales Representative           | Obere Str. 57       |
| Berlin NULL 12209              | Germany             |
| 030-0074321 030-0076545        |                     |
| BLAUS Blauer See Delikatessen  | Hanna               |
| Moos Sales Representative      |                     |
| Forsterstr. 57                 | Mannheim NULL       |
| 68306 Germany                  | 0621-08460          |
| 08924                          | 0 6 2 1 -           |
| DRACD Drachenblut Delikatessen | Sven                |
| Ottlieb                        | Order Administrator |
| Walserweg 21                   | Aachen NULL 52066   |
| Germany                        | 0241-039123         |
| 0241-059428                    |                     |
| ...                            |                     |



Microsoft  
BackOffice

Hogy került oda az a csillag? Ez a lusta SQL programozók egyik legnagyobb öröme (és a Query Optimizer egyik legnagyobb bánata). Ha nem akarunk minden oszlopot kézzel felsorolni a SELECT után, akkor használhatjuk a csillagot, ami kilitáz minden oszlopot. Csak bánjunk vele csinján! Gyorsak ugyan a mai számítógépek, de amikor tízezres nagyságrendű sorok mellett egyenként is kilobájtos méretű sorokat válogatunk le, hát bizony bámulhatjuk a homokórárt, vagy az időtűllépésről szóló hibaiüzeneteket, nem is beszélve a haragos ügyfélről.

A WHERE záradék után megadott logikai feltételekkel szűkíthetjük a listázás eredményhalmazát (Result Set). Mi csak azokat a sorokat listáztuk, amelyekben a Country attribútum (oszlop) értéke megegyezik a „Germany” szöveggel. Itt tetszőlegesen bonyolult logikai kifejezések állhatnak, csak igyekezzünk jól tagolni, hogy pár hónap múlva is olvasható és érthető legyen a korábban zseniálisnak kikiáltott feltételünk. Például:

```
SELECT
  CompanyName, Country, Fax
FROM
  Customers
WHERE
  Country = 'Germany' AND
  Fax IS NULL
```

|                   |         |      |
|-------------------|---------|------|
| Königlich Essen   | Germany | NULL |
| Morgenstern Ge... | Germany | NULL |
| QUICK-Stop        | Germany | NULL |

Azaz kérünk minden olyan sort, ahol a cégnev Germany, és a Fax mező értéke NULL. De mi az a NULL? Ezt a kérdést sokszor többéves tapasztalattal rendelkező SQL programozók sem tudják, pedig igen lényeges a tisztánlátás az ő tulajdonságait illetően.

A NULL egy SQL adatbázisban azt jelenti, hogy nincs adat. Nem keverendő össze a 0 számmal, vagy az „” üres sztringgel, még kevésbé a NULL pointerrel! Majd az illesztéseknél (JOIN) látjuk, hogy igen sok problémát okozhatnak, úgy-hogy szemmel kell tartani őket. Látható, hogy az összehasonlító operátor (is) más, mint amit egyéb, „normális” adatra alkalmazunk (=, <, > stb.). A nyelv szerzői ezzel is meg akarták különböztetni a NULL-t a valós adatoktól. Hisz 2 > NULL az igaz? Hmm. A jó ég tudja. Pont ezért problémás a NULL adatok kezelése, mert amíg normális adatokra megfelelően működnek az operátorok, NULL-ra nem. Ez nem jelenti azt, hogy hibát jeleznek, vagy véletlenszerűen viselkednek. Természetesen következetes módon fognak viselkedni, csak meg kell nézni a kézikönyvben, hogy milyen feltételek esetén milyen eredményt adnak, vagy ami ennél szászor jobb megoldás: ahol csak lehet, kerüljük el a NULL-okat. Ingyencenek: a Microsoft SQL Server fizikai adattárolási struktúrája olyan, hogy minden egyes NULL engedélyezett attribútumhoz van egy jelzőbit, amely jelzi, hogy tartalmaz-e a mező tényleges adatot, vagy nem (azaz NULL). Ez a sort leíró fejlécben van, hisz ezt minden sor minden NULL-ozható attribútumára tárolni kell. Azaz ez a plusz adminisztráció még egy kis teljesítményhátrányt is jelent.

#### Csak számolunk, csak számolunk

Hagyjuk a ronda NULL-okat, és haladjunk tovább a normális adatok világában. Mi van, ha a big boss abc sorrendben szeretné látni az amerikai megrendelőket?

```
SELECT
  CompanyName
FROM
  Customers
WHERE
  Country = 'USA'
ORDER BY
  CompanyName
```

```
Great Lakes Food Market
Hungry Coyote Import Store
Lazy K Kountry Store
Let's Stop N Shop
...
```

F5, és nagy az öröm. Az ORDER BY után több mezőt is fel lehet sorolni, azaz, hogy először az első mező szerint rendezzen sorba, majd ezekben a csoportokon belül rendezze tovább a második, stb. attribútum szerint. A rendező mezőnevek közé vesszőt kell rakni, hasonlóan, mint a SELECT után. Pl.:

```
SELECT
  Country, City, CompanyName,
  ContactName
FROM
  Customers
ORDER BY
  Country, City, CompanyName
```

A főnök kíváncsisága csillapíthatatlan. Kíváncsi rá, hogy meddig ér a lepedője, azaz mely országokkal van kapcsolata a cégének. Am legyen:

```
SELECT DISTINCT
  Country
FROM
  Customers
```

```
Argentina
Austria
Belgium
Brazil
...
```

Mit is mondtunk az SQL Server-nek? Listázza ki az összes különböző (DISTINCT) országot a Customers táblából. Ez olyan egyszerű SQL-ben, mint ahogyan látszik! Aki hozzászokott a procedurális gondolkodáshoz, annak szokatlan ez a fajta gondolkodásmód, ahogyan SQL nyelven definiáljuk a problémákat. Egy procedurális nyelven (C, Pascal, Basic stb.) úgy válogatnánk le a sorokat, hogy végigmennénk minden soron, egybegyűjtենék a már megtalált országokat, és minden egyes még feldolgozatlan sornál megnéznénk, hogy már benne van-e az adott tétel a megtaláltak között. Ha nem, felvesszük, ha igen, akkor csak egyszerűen továbblépünk a következőre. Azaz procedurális esetben sorokban, egyedi adatokban gondolkodunk, míg a másik

esetben halmazokban. Több év C által megfertőzött szekenciális gondolkodás után szokatlan lehet ez, de könnyen meg lehet szokni, és miután valami ráérett az izére, rájön, hogy ilyen módon sokkal egyszerűbben és tömörebben meg lehet fogalmazni a problémákat (*nem véletlen, hogy így módon fejlesztették ki az SQL nyelvet*). Másrészt kurzorok használatában át lehet térni szekenciális feldolgozásra, de erről majd egy teljes cikk fog szólni. Na, de félre az ideológiákkal, számoljuk meg, hány ügyfélünk van Franciaországban:

```
SELECT
COUNT(*)
FROM
Customers
WHERE
Country = 'France'

11
```

Magyarul: számoljuk meg az összes sort, ahol az ország oszlopban Franciaország áll. A COUNT az aggregáló függvények egyik jeles képviselője. Ezeknek a függvényeknek az a közös jellemzőjük, hogy több sorból képeznek valamilyen végeredményt, a sorok valamelyik attribútumát felhasználva. Ebből a szempontból a COUNT(\*) egy kicsit speciális, mert az oszloptól függetlenül egyszerűen megszámlálja a sorokat. Egy újabb példán keresztül nézzünk egy kicsit mögé a COUNT függvénynek.

Hány cégnek van faxa, vagy legalábbis úgy tudjuk róla, hogy van neki (*azaz a Fax mező nem NULL*)?

Klasszikus megoldás:

```
SELECT
COUNT(*)
FROM
Customers
WHERE
Fax IS NOT NULL

69
```

Microsoft SQL Server specifikus, de jól működő megoldás:

```
SELECT
COUNT(Fax)
FROM
Customers

69
```

Miért működik ez jól? Miért nem számlolja meg az összes sort, miért csak azokat, amelyekben a Fax mező nem NULL? Mert így logikus. Mivel a NULL azt jelenti, hogy nincs adat, vagy nem tudunk róla semmit, ezért nem is szabad bevenni az ilyen számlálásokba. Ez megint csak a NULL értékek sajátossága (*mondtam, hogy sok baj van a NULL-okkal*). Az összes aggregáló függvény bokorugrásban megy tovább a következő sorra, ha a feldolgozás alatt álló mezőben NULL van. Hisz mit kezdene egy átlagszámító függvény a NULL-lal?

A COUNT(\*) azért lóg ki a sorból, mert minden sort beszámol, még akkor is, ha minden attribútum értéke NULL. A két megoldás egyformán hatékony, de míg az előbbi jól olvasható és nem tartalmaz implicit megállapodásokat, ad-

dig a második néhány hónappal a fejlesztés után már fejtorést okozhat, hogy mi a csudát akartam én kezdeni azzal a kifejezéssel. Azaz nem javasolom a második megoldást. Aki úgy érzi, hogy a második megoldás gyorsabb, annak elmondom, hogy az SQL Server esetén szintaktikai bravúrokkal általában nem sikerül teljesítményt javítani, mert úgyis azzal kezd az utastások végrehajtását, hogy lebontja őket elemi részekre, és az *(átlala vélt)* legoptimálisabb módon fogja végrehajtani. Eddig még legtöbbször okosabb volt nálam :) Maradjunk még egy kicsit az aggregáló függvényeknél! További tipikus példák a MIN, a MAX és az AVG és a SUM. Az első kettő egyértelmű, az AVG számtani közepet (*népi nyelven átlag*) számol, a SUM pedig összeadja a megadott mezőket. Nézzünk egy példát, amiben felhasználjuk mind egyiket! Keressük meg a legkisebb és a legnagyobb egy-egységű termék árát, az egységárak átlagát és az egységárak összegét az összes termékre vonatkozóan.

```
SELECT
MIN(UnitPrice),
MAX(UnitPrice),
AVG(UnitPrice),
SUM(UnitPrice)
FROM
Products

2.5000      263.5000      28.8663      2222.7100
```

### Veszedelemes viszonyok

Vegyünk nagy levegőt, és mélyedjünk el egy kicsit az adatbáziskezelés elméletébe. Miért mondják az SQL Serverre, hogy relációs adatbáziskezelő? Azért, mert a benne levő entitások (*ennek magyar megfelelője az izé, de fizikai adatbázisban táblának felel meg*) között logikai kapcsolatokat, relációkat fogalmazhatunk meg. Ezt azért találták ki, mert így az ismétlődő adatokat kiemelhetjük külön táblákba, egyrészt helyet spórolva meg, másrészt az adatbázis konzisztenciájának biztosítása végett. Lássunk erre egy példát. Az Orders tábla tárolja az ügyfelek vásárlásait. Azonban minden egyes sorban az ügyfelet csak egy azonosító jelzi (*pl.: RATTC*), amely azonosító alatt futó ügyfél valódi adatai a Customers táblában vannak. Ezzel helyet spóroltak, hisz nem kell többször leírni az ügyfél címét, nevét stb. minden egyes megrendelésnél, másrészt nem fogunk találni olyan sorokat, hogy megrendelő Egér Béla, a másikban, hogy Eger Béla és így tovább. Így az adatbázis logikailag konzisztens marad, hisz ugyanarra a valóságos egyedpéldányra (*Eger Béla*) nem hivatkozhatunk többféleképpen. Egyébként az a folyamatot, amikor a redundáns részeket kirakjuk külön táblába, normalizálásnak nevezzük. Ebből következően azok az adatbázisok normálisak, amelyekben sok, de keskeny, azaz kevés oszlopot tartalmazó tábla van. Az elméleti maximumig agyonnormalizált adatbázist azonban keveset látnak a világban. Ennek két oka van. Egyrészt sokan nem is tudják, hogyan kell normalizálni (*és hogy egyáltalán kell*), másrészt teljesítményemegfontolások miatt sokszor normalizálás után denormalizáljuk valamilyest az adatbázisunkat. Egy nem megfelelően normalizált adatbázissal az a baj, hogy nehéz a konzisztenciáját megtartani. Például egy oszlopban tároljuk egy termék árát, egy másikban az ÁFA-ját, egy harmadikban pedig az ÁFA-s árat. Ez gyönyörű példája a NEM normalizált adatbázisnak (*triviális függőség van az oszlopok között*). A baj akkor kezdődik, amikor megváltozik a termék ára, és elfelejtjük módosítani az ÁFA-s árat. Egy





Microsoft  
BackOffice

hét (perc) múlva, amikor már senki nem emlékszik, melyik mező volt módosítva, melyik árat fogadjuk el helyesnek? Újra együtt!

Vissza a gyakorlatba, hogyan lehet összehozni a szétdarabolt információkat? Ehhez lesz szükségünk a JOIN kulcszóra. Listázzuk ki a megrendelők által kért megrendelések dátumát és a szállítás városát:

```
SELECT
  CompanyName, OrderDate, ShipCity
FROM
  Customers
INNER JOIN
  Orders
ON
  Customers.CustomerID =
  Orders.CustomerID

CompanyName OrderDate ShipCity
Vins et alcools Chevalier 1996-07-04
Reims
Toms Szpecialitáten 1996-07-05 Münster
Hanari Carnes 1996-07-08 Rio de Janeiro
...
```

Azaz a Customers táblát illessze az Orders táblához, mégpedig azokon a pontokon, ahol a Customers tábla CustomerID mezője megegyezik az Orders tábla CustomerID mezőjével. Minden egyes egyezésként készítek egy „hosszú” sort, azaz egymás mellé rakja a két tábla összeillesztett sorát, amelyekből mi csak a SELECT után felsorolt oszlopokat kerjük. SQL Server 6.5-ig ez úgy ment, hogy vette az első tábla első sorát, és megnézte, hogy az ON után megadott feltételt alkalmazásával a másik táblában talál-e egy párt a vizsgált sornak. Ha igen, akkor egymás mellé illesztette őket, letárolta, és folytatta a második tábla következő sorával, hisz általában több sor is illeszkedhet a vizsgált sorhoz. Ez első ránézésre hihetetlen lassú folyamat, hisz, a két tábla sorai számának szorzata adja az összehasonító műveletek számát. Az indexek használata miatt ez szerencsére nem így van. Ennek az illesztési eljárásnak a (jól eltalált) neve Nested Loop Join. SQL 7-től még két további illesztő algoritmus áll rendelkezésre, amelyeket nagyon sok sort adó lekérdezéseknél szeret választani (Hash és Merge Join).

A feladat jellegének megfelelően tetszőleges számú táblát össze lehet kapcsolni. PL:

```
SELECT
  CompanyName, OrderDate,
  ProductName, Products.ProductID
FROM
  Customers
INNER JOIN
  Orders
ON
  Customers.CustomerID =
  Orders.CustomerID
INNER JOIN
  [Order Details]
ON
  Orders.OrderID = [Order
  Details].OrderID
INNER JOIN
```

```
Products
ON [Order Details].ProductID =
Products.ProductID

CompanyName OrderDate ProdName
ProdID
Blondesd... 1996-07-25 Alice Mutton 17
Lehmans... 1996-08-13 Alice Mutton 17
Rattlesn... 1996-08-30 Alice Mutton 17
```

Két dolgot figyeljünk meg a fenti lekérdezésben! Ha olyan oszlopot listázzunk ki, amelynek neve több táblában is szerepel (pl.: ProductID, OrderID, CustomerID), akkor meg kell jelölni, hogy melyik táblából akarjuk kilistázni az adatokat. Ez INNER JOIN-nál még mindegy lenne, de a többi JOIN-nál nagyon fontos lesz. A másik lényeges pont, hogy a szóközt is tartalmazó tábla- és oszlopneveket szögletes zárójelek [] közé kell tenni. Ugyanez érvényes, ha kulcszót akarunk használni tábla- vagy attribútumnévnek. Where nevű tábla nem túl gyakori, de a User tábla már annál inkább, ami pedig kulcszó...

Az INNER JOIN működéséből következők, hogy az összes olyan sor kimarad az eredményből, amelynek nincs párja a másik táblában. Azonban a gyakorlatban sokszor az árva sorokra is szükség van. A kettővel ezelőtti példánál maradvakat a vásárlókat is ki szeretnénk listázni, akiknek (még) nincsenek megrendeléseik. Ekkor jön segítségünkre az OUTER JOIN. Ennek két alfaja van, a LEFT OUTER JOIN és a RIGHT OUTER JOIN. Mivel a JOIN kulcszó mindig két tábla között helyezkedik el, a jobb és bal irány értelmezése természetesen adódik. Amelyik táblát így kitüntetjük, abból az összes sor kilistázzódik, azok is, amelyeknek nincs párja a másik oldali táblában. Így a példánk:

```
SELECT
  CompanyName, OrderDate, ShipCity
FROM
  Customers
LEFT OUTER JOIN
  Orders
ON
  Customers.CustomerID = Orders.CustomerID

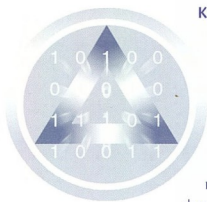
Rattlesnake Canyon 1998-05-06 Albuquerque
Paris spECIALITÉS NULL NULL
FISSA Fabrica NULL NULL
```

Ahol az ügyfélhez nincs megrendelés az Orders táblában, ott a kiszolgáló NULL-t helyez el azokban az oszlopokban, amelyek az Orders táblára mutatnak. Ez egybevág a NULL azon jelentésével, hogy „nincs adat”.

Sajnos ezen a ponton be kell fejezzem barangolásomat a lekérdezések birodalmában, de ne aggódjanak, a következő számban tovább túrunk-fúrunk a SELECT rejtelmeiben, hogy azután rátérhessünk az igazi izgalmakra, az adatok módosítására.

Szócó Zsolt, MCSE  
Protomix Kft.





### Kedves Olvasóink!

Az alábbi cikk kísérleti jelleggel szerepel lapunkban: szeretnénk megmutatni, hogy az Internet életét meghatározó szabványok (RFC, *Request For Comments*), melyekkel minden komoly rendszergazda előbb-utóbb szembesül valójában egyáltalán nem száraz, tudományos értekezések, hanem érdekes módon görődülékeny, olvasmányos dokumentumok, melyeket akár még kikapcsolódásképpen is élmény kézbe venni. A sorozat szándékunk szerint kéthavonta jelentkezne, s mindig egy-egy teljes RFC fordítását tartalmazza majd. Kérjük észrevételeit, a rovat hasznos vagy haszontalan voltát a [Tech.Net@lyris.netacademia.net](mailto:Tech.Net@lyris.netacademia.net) címen található levelezési listán írja meg nekünk!



Köszönettel  
A főszerkesztő

És innentől következen a 1597-es RFC teljes szövege, melyet azért választottunk, mert a fenti ígéretesnek tökéletesen megfelel: olvasmányos, és nem túl bonyolult!

Network Working Group  
Request for Comments: 1597

Y. Rekhter

T.J. Watson Research Center, IBM Corp.

Kategória: Informatív

B. Moskowitz  
Chrysler Corp.  
D. Karrenberg  
RIPE NCC  
G. de Groot  
RIPE NCC  
1994. március

### RFC 1597: Cím kiosztás magáninternetek számára

#### E feljegyzés célja

Ennek a feljegyzésnek az a célja, hogy információkat nyújtson az Internet közössége számára, de nem határoz meg semmiféle szabványt, és korlátozás nélkül terjeszthető.

#### 1. Bevezetés

Ebben az RFC-ben olyan módszereket teszünk közzé, melyek az IP címter gazdaságos kihasználását teszik lehetővé azért, hogy a magánhálózatok számítógépei számára – melyeknek nem szükséges külső kapcsolatokat fenntartani – nem igényelünk globális Internetcímeket. Természetesen ezek a módszerek továbbra is lehetővé teszik a magánhálózaton belüli kapcsolatokat, sőt a megfelelő gépek külső Internetkapcsolatait is. A szerzők azt remélik, hogy ezeket használva jelentős megtakarítások érhetők el az egyre fogyó IP címek kiosztásában.

Ebben a feljegyzésben a „vállalat” szót a továbbiakban egy különálló TCP/IP-alapú hálózatot üzemeltető entitásként határozzuk meg, amely a magánhálózat cím kiosztásának részleteit is meghatározza.

#### 2. Elérendő célok

A TCP/IP-alapú hálózatok világszerte bekövetkezett elterjedésével – az Interneten kívülre is beleértve – egyre növekvő számú olyan hálózat létrejöttét figyelhetjük meg, melyek egyáltalán nincsenek az Internethez csatlakoztatva. Ezek a hálózatok csupán a belső számítógépek összeköttetésére használják a TCP/IP protokollt. Sok vállalat a jövőben sem fogja magánhálózatát más vállalati hálózatokkal, vagy az Internettel összekapcsolni.

A jelenlegi gyakorlat szerint minden kiszolgáló, mely TCP/IP-t használ, globálisan egyedi címet kap. Am egyre többen fejezik ki aggodalmukat amiatt, hogy a véges IP-címter egyszer csak kimerül. Éppen ezért az IP címterek hozzárendelésére vonatkozó irányelveket jelentősen megszigorították az utóbbi években [1]. Ezek a szabályok konzervatívabbak, mint azt a vállalatok – hálózataik megvalósításához és működtetéséhez – szeretnék.

A magánhálózatokon használt, IP-t használó számítógépek három kategóriába sorolhatók:

- ☞ Olyanok, melyeknek nem szükséges más vállalati hálózatok gépeivel vagy az Internettel összeköttetésben állni.
- ☞ Olyan gépek, melyeknek a külső szolgáltatások (például: elektronikus levelezés, FTP, hírszolgáltatás, távoli bejelentkezés) korlátozott részhalmozának elérésére van szükségük. Ezt alkalmazásszintű átjárókkal is meg lehet oldani.
- ☞ Olyan gépek, melyeknek a vállalati hálózaton kívüli, IP-n keresztüli hálózati elérésre van szükségük.

Az első kategóriába eső számítógépek olyan címeket használhatnak, melyek a vállalati hálózaton belül egyediek, tágabb értelemben (vállalatok között) azonban nem.

A legtöbb második kategóriába eső számítógép számára a korlátozás nélküli külső (IP-n keresztüli) kapcsolat nemhogy szükséges, hanem – biztonsági okokból – egyenesen kerüendő is. Ezek a gépek – az első kategóriába esőkhöz hasonlóan – olyan címeket használhatnak, melyek csak a magánhálózaton belül számitanak egyedieknek, de azon kívül nem.

Csak az utolsó kategóriába eső gépeknek van szükségük globálisan egyedi címekre.

A legtöbb alkalmazásnak csak a vállalaton belüli kapcsolatokra van szüksége, és csak nagyon kevés külső géppel kell kapcsolat tartania. A nagyobb vállalatoknál könnyen meghatározható, hogy mely gépeknek van szükségük külső hálózati kapcsolatra.

Néhány példa azokra az esetekre, ahol a külső kapcsolat valószínűleg nem szükséges:

- ☞ Repülőter, ahol az érkezési és indulási adatokat kijelző táblák külön címezhetők TCP/IP segítségével. Valószínű, hogy ezeket a kijelzőket más hálózatok számára elérhetővé kellene tenni.
- ☞ Nagykereskedelmi hálózattal rendelkező szervezetek – például bankok és kereskedelmi értékesítési láncok – melyek TCP/IP-re cseréltek belső kommunikációs hálózatukat. Itt nagyszámú munkaállomás létezik, például pénztárgépek, pénzkidő automaták és egyéb irodai gépek, melyeknek aligha szükséges a külső kapcsolat.
- ☞ Biztonsági okokból számos vállalat használ alkalmazásszintű átjárókat (például tűzfalakat) belső hálózatának



Internetre csatlakoztatására. A belső hálózatnak általában nincs közvetlen Internetelérése, így csak egy vagy több tűzfalnak kell láthatónak lennie az Internetről. Ebben az esetben a belső hálózat nem egyedi IP-címeket is használhat.

☞ Ha két vállalati hálózat saját kapcsolatain keresztül kommunikál egymással, akkor a kapcsolaton át a gépeknek csak egy nagyon korlátozott része érhető el kölcsönösen. Csak ezeknek a gépeknek van szüksége egyedi IP címekre.

☞ Általában szükséges, hogy a belső hálózat útválasztóinak felülete közvetlenül elérhető legyen a vállalati hálózaton kívülről.

### 3. A privát címtér

Az Internet Assigned Numbers Authority (IANA) az IP-címtér következő három blokkját foglalta le a magánhálózatok számára:

**10.0.0.0 – 10.255.255.255**  
**172.16.0.0 – 172.31.255.255**  
**192.168.0.0 – 192.168.255.255**

Az első blokkra „24 bites blokk”, a másodikra „20 bites blokk” a harmadikra pedig mint „16 bites blokk” fogunk hivatkozni. Vegyük észre, hogy az első blokk nem más, mint egy A-osztályú hálózat címe, a második blokk 16, egymás után következő B-osztályú hálózat címeinek összessége, míg a harmadik blokk 255 egymást követő C-osztályú hálózat IP címeinek halmaza.

A dokumentumban definiált IP címtereken kívüli címek is használhatók (azaz az IANA vagy más Internetcím kiosztó szervezet által ajánlottakon kívül is választhatunk belső IP címeket), de ezek a címek csak a privát hálózatban lesznek használhatóak. Ha egy vállalatnak globális egyedi címtérre van szüksége, Internetcím kiosztó szervezettel kell igényelni azokat. Az így kapott címtartomány sohasem esik a fent definiált tartományokba.

A privát címtér használatához el kell döntenünk, hogy – belátható időn belül – mely számítógépeknek van szükségük külső kapcsolatra, és melyeknek nem. Ez utóbbiakat – melyek a fent meghatározott privát címteret használják – nevezhetjük privát számítógépeknek. A privát gépek a magánhálózat bármely más – akár privát, akár nyilvános – gépével kommunikálhatnak, de nem tudnak külső gépekkel IP kapcsolatot létesíteni.

A privát gépek annak ellenére, hogy nem rendelkeznek külső kapcsolattal, képesek külső szolgáltatásokat igénybe venni – alkalmazásszintű átjárók segítségével.

Minden más számítógépet nyilvánosnak hívunk, mert ezek globálisan egyedi címtér-hozzárendeléssel rendelkeznek egy Internetcím kiosztó szervezet jóvoltából. A nyilvános gépek a hálózaton belül mind a privát, mind a nyilvános gépekkel kommunikálhatnak, valamint külső gépekkel IP-alapú kapcsolatot is fenntarthatnak. A nyilvános gépek más vállalati hálózatok privát gépeivel nem léphetnek kapcsolatba. Egy gép privát státusának nyilvánossá változtatása IP címének megváltoztatásával történik. Mivel a pri-

vát címeknek nincs globális jelentésük, az útválasztási információt nem szabad az egyes vállalati hálózatokat összekötő kapcsolatokon át terjeszteni, valamint a belső hálózati címre vonatkozó forrás vagy célcímmel rendelkező csomagokat sem szabad a hálózaton kívüire továbbítani. Elvárható, hogy azok az útválasztók, melyek nem használják a privát címteret (különösen azok, melyeket az *Internetszolgáltatók üzemeltetnek*), ne vegyék figyelembe (szűrjék ki) a magánhálózatokról származó esetleges csomagokat és útválasztási információkat. Az útválasztásnak az ilyen információk kiszűrését nem szabad az útválasztási protokoll hibájáért kezelnie.

Az ilyen címekre vonatkozó indirekt hivatkozások a hálózaton belül kell tartani. Ilyenek például a DNS erőforrás-rekordok és más, a magánhálózat belső privát címére vonatkozó információk. A gyakorlatban az Internetszolgáltatóknak kell megtenniük a megfelelő intézkedéseket, hogy a fenti problémákat elkerülhessük.

### 4. A privát címtér használatának előnyei és hátrányai

Az Internet számára nyilvánvaló előny, hogy a privát címtér használata megkíméli a globálisan egyedi címteret, hiszen azokban az esetekben, ahol nem szükséges, nem használja azt. A vállalatok is sok előnyét élvezhetik a privát címtér használatának: sokkal rugalmasabban alakíthatják ki hálózataikat a rendelkezésükre álló nagyobb címtér segítségével, mint ahogy azt a globálisan egyedi címek használatával tehetnék. Egyúttal lehetővé válik számukra, hogy saját működtetési és felügyeleti igényeiknek megfelelően osszák ki a címeket és könnyebben bővíthessék hálózataikat.

Az Interneten – különböző okokból – már történtek olyan esetek, melyeknél a korábban Internethez nem kapcsolt vállalat az IANA-val nem egyeztetett címteret használt gépeihez. Néhány esetben ezt a címteret már más vállalatok lefoglalták. Amikor eljön az idő, hogy egy ilyen vállalat az Internethez kapcsolódjon, súlyos problémákkal kell szembenéznie, mivel a többértelmű címzés miatt az IP útválasztás nem fog megfelelően működni.

A privát címtér használata elkerülhetővé teszi a címzési problémákat, s ezzel együtt biztosítja a külső kapcsolattartás lehetőségét is.

Persze lehet azon vitatkozni, hogy a címváltoztatás mindig fennálló lehetősége vajon jelentős hátrányt jelent-e, ha a hálózati címeket a nem magán Internetek számára kiosztott címtérből választjuk. Azt kell megvizsgálnunk, hogy ez a lehetőség valóban oly fenyegető-e, hiszen sok olyan számítógépet használnak, melyet sohasem kell átSOROLNI a harmadik kategóriába. Olyan hálózatok is szép számmal akadnak, melyeket sohasem fognak IP-vel összekapcsolni más hálózatokkal. De még ha a címváltoztatás elkerülhető is (a *Classless Inter-Domain Routing-gal: CIDR*) a vállalati hálózat, mely Internethez van csatlakoztatva, Internet szolgáltatójának megváltoztatása esetén rákényszerülhet nyilvános gépei át címezésére is. Ez a fajta át címezés kényszer a jövőben valószínűleg gyakrabban lép majd fel, függetlenül attól, hogy a vállalat a magánhálózatok számára fenntartott címtérből gazdálkodik-e vagy sem. Segédeszközök használatával (például DHCP) csökkenthetjük a probléma által okozott fejfájást.



Azt is figyelembe kell vennünk, hogy a nyilvános és privát gépek világos elkülönítésével a felügyelet nélküli külső elérés kockázata csökkeni fog, így a tervezésbe fektetett többletmunkát akár biztonsági intézkedésként is felfoghatjuk.

### 5. Üzemeltetési megfontolások

Az ajánlott stratégia szerint először tervezzük meg a hálózat kívül-világ elől elzárt részét, a privát címeteret használva a belső kapcsolatokhoz. Ezután kerítünk sort a nyílt alhálózatok tervezésére, végül a külső kapcsolatokat határozzuk meg. Ez nem a végleges terv. Ha bizonyos számú számítógép státuszát kell megváltoztatnunk, csak a kérdéses gépeket kell átneveznünk és ha szükséges, másik fizikai alhálózatot kell kiépítenünk.

Ha megfelelő alhálózati séma tervezhető, és a használatba vevendő eszközeink támogatják, ajánlatos a privát címter 24 bites blokkját felhasználni és a címkiórási tervbe a hálózat növekedését is beleszámítani. Ha az alhálózatok létrehozása problémát jelent, használjuk a 16 bites C-osztályú blokkot, mely 255 egymás után következő C-osztályú hálózat címét tartalmazza.

Több IP (a)hálózat használata egyazon fizikai médiumon számos veszélyt rejt magában. Azt ajánljuk, hogy ha csak tehetjük, kerüljük el ezt a megoldást, kivéve ha a működtetésben rejlő problémákat teljes mértékben megértjük, és minden eszközünk képes ennek az üzemmódnak megfelelően működni.

Egy számítógép státuszának megváltoztatása (*privától nyilvánosra*) maga után vonja címének, és a legtöbb esetben fizikai kapcsolódásának megváltoztatását is. Olyan helyeken, ahol ez a fajta változás előre látható (*géptemek*) ajánlatos elkülönített fizikai hálózatot használni a nyilvános és a privát alhálózat számára.

Az összes számítógép státuszának megváltoztatása az egész (a)hálózaton egyszerűen végezhető el, anélkül, hogy az egész vállalati hálózat működését fel kellene függeszteni. Ebből következik, hogy ajánlatos csoportba foglalni azokat a számítógépeket, melyek csatlakozási igényei a jövőben hasonló változásokon fognak keresztül menni saját alhálózatukon.

Azoknál az útválasztóknál, melyek a vállalati hálózatot kapcsolják össze a külső hálózattal, nyomatékosan ajánlott, hogy a kapcsolat mindkét végén a megfelelő csomag- és útválasztósűrőket állítsuk be, hogy megelőzzük a csomagto-vábbítás és útválasztási információk okozta problémákat. A vállalati hálózatnak ki kell szűrnie az összes külső magánhálózat-tól származó útválasztási információt, hogy megvédje magát a nem egyértelmű útválasztási helyzetektől, melyek akkor léphetnek fel, ha a privát címter a vállalati hálózaton kívülre mutat. Olyan csoportok, melyek előre látják kölcsönös kommunikációs igényeiket elhatározhatják, hogy közös címkiórási terv felállításával közös hálózatot hoznak létre, melyet a szükséges szervezeti rendelkezések – például nyilvántartás – bevezetésével is támogathatnak.

Ha a vállalat két különböző pontja között külső szolgáltató bevonásával szükséges összeköttetést létesíteni, akkor számításba lehet venni valamilyen IP-alagút használatát, mellyel megelőzhető a csomagok elvesztése.

A DNS bejegyzések elvesztésének elkerüléséhez két névki-szolgáló működtetése ajánlott. Egy külső kiszolgáló felelős a vállalati hálózat minden globálisan egyedi IP címéért, a másik a belső névki-szolgáló, mely a vállalati hálózat minden –

külső és belső – IP-címéért felel. Az egységesség érdekében állítsuk be mindkét névki-szolgálót ugyanazon adatok alapján úgy, hogy a külső kiszolgálót csak a számára szükséges DNS-információkkal lássuk el.

Minden belső számítógépen lévő (*belső vagy külső névre vonatkozó kérelmet kiszolgáló*) névfeloldó csak a belső névki-szolgálót kérdezze le. A külső névki-szolgáló a vállalati hálózaton kívüli, külső kérésekkel foglalkozzon, és a globális DNS-hez legyen kapcsolva. A belső névki-szolgáló minden, a vállalati hálózaton kívüire mutató információkérést irányítson át a külső névki-szolgálóra. Ezzel az elrendezéssel a belső gépek hozzáférhetnek a globális DNS információhoz, ám a belső gépekről szóló információk nem jutnak el a vállalati hálózaton kívül működő számítógépekhez.

### 6. Referenciák

[1] Gerich, E., „Guidelines for Management of IP Address Space”. RFC 1466, Merit Network, Inc., 1993. május

### 7. Biztonsági megfontolások

Bár a privát címter használata növelheti a biztonságot, nem helyettesítheti a megfelelő biztonsági eszközök és intézkedések használatát.

### 8. Zárzó

A leírt séma segítségével számos nagyvállalat a globális IP címter viszonylag kevés címblokkjának felhasználásával oldhatja meg elérhetőségének problémáját. Az eljárás használata a globális IP címter élettartalmának kitolódása révén jelentkezik. A vállalatok a viszonylag nagykiterjedésű privát címter rugalmasságát fordíthatják előnyükre.

## Microsoft



Elkészült a .net platform első hírdője, a BackOffice programcsalád új változata, új nevén: .Net Enterprise Servers. A BackOffice kiszolgálócsalád tagjainak legújabb változatait tartalmazó programcsomagot a szeptember 26-i nemzetközi bemutatók után október 12-én jelentik be Magyarországon. Lapunk különszámmal készült az eseményre, amelyben sorra bemutatjuk a család tagjait: ezek az Exchange 2000 Server, az SQL Server 2000, az SMS Server 2000, a Host Integration Server 2000 (korábban SNA Server), az ISA Server 2000 (korábban Proxy Server) és természetesen a Windows 2000 Server SP1, valamint sok más „kisebb” összetevő. A Exchange 2000 Server 120 napos próbaverziója a <http://www.microsoft.com/exchange/productinfo/Eval.htm>, az ISA Server legújabb bétája (RC1 – a cikk írásának pillanatában még elérhető) pedig a <http://www.microsoft.com/isaserver/productinfo/evaedition.htm> címről tölthető le.

### Várható Whistler időrend

A Windows SuperSite-on (<http://www.winsupersite.com>) olvasható rengeteg Whistler-információ között megbújik a Microsoft új operációs rendszere megjelenésének várható időrendje is. Ezek szerint a Whistler Beta 1 október közepe táján, a Beta 2 valamikor december elején, a majdnem késznek mondható RC1 várhatóan 2001 februárjában jelenik meg. Ezt követi majd – körülbelül egy hónap múlva – az RC2, majd – ha minden jól megy – az RTM (Released To Manufacture, gyártásra kész) változat, mégpedig állítólag 2001. április 18-án. (Lapzártaok ez a terv máris két hetet csúszni látszik...)

A Whistler a .net architektúra alapját képezi (mostanában Windows .NET 1.0-nak is nevezik), természetesen számos látványos és hatékony újítással. Az operációs rendszer Windows 2000 alapokra épül, a Windows 9x/Me vonal ezzel megszűnni látszik. A Whistler aktuális verziójának újdonságait (többek között az új, HTML-alapú felhasználói felületet) áttekintő cikk a [http://www.winsupersite.com/reviews/whistler\\_2257.asp](http://www.winsupersite.com/reviews/whistler_2257.asp) címen található, rengeteg képernyőképpel meghintve.



### 25 éves a Microsoft

A jövő után tekintünk egy pillanatra vissza a múltba: a Microsoft 1975-ös megalakulása után idén ünnepli 25. születésnapját.



A Microsoft múzeum weblapján (igen, a múzeum a valóságban is létezik :-)) a <http://www.microsoft.com/mscorp/museum/anniversary.asp> címen betekinthetünk a Microsoft életének elmúlt negyed századába.

### Active Directory Client for Windows NT 4.0 Workstation

A Windows 2000 megjelenésekor sokan fúcskáltak, hogy míg a Windows 95/98-ra készült Active Directory Client (ami – ha jól tudjuk – Windows Me-n egyébként nem működik), a Windows NT 4.0-hoz való példány hiányzott a Windows 2000 CD-ről.

Az Active Directory Client for Windows NT 4.0 Workstation béta változata most letölthető a <http://www.microsoft.com/windows2000/adclients/> oldalról, a végleges változat a Service Pack 7-be fog bekerülni.



Jó hír a fejlesztőknek: itt a Visual Studio.Net Beta 1 A <http://msdn.microsoft.com/vstudio/nextgen/betaone.asp> cím minden fejlesztőnek kötelező olvasmány. Mire az oldalon található információkon átrágjuk magunkat, az MSDN előfizetésben valószínűleg megérkezik a Visual Studio új generációjának béta változata.

A fejlesztőkörnyezet a továbbfejlesztett Visual C++, Visual Basic és Visual FoxPro mellett természetesen telis-tele lesz .net elemekkel: itt az #, az ADD plus, a Windows Forms, XML és SOAP támogatás, minden, amit a .net fejlesztő szem-szája csak megkívánhat!



### Végül, egy „személyes” jellegű hír: Indul előfizetési akciónk!

Ha szeretné, hogy magazinunk minden hónapban biztosan megjelenjen postáládájában, fizessen elő! Az előfizetési akciónk érdeklődjön a oldalunkon, a

<http://technet.netacademia.net> címen!





### IIS5 és az NTFS: jogosultsági ütvésztk Hogyan működik az NTFS jogosultsági rendszere?

Ha a rendszergazda létrehoz egy felhasználói fiókot, akkor a Windows ehhez automatikusan hozzárendel egy egyedi, biztonsági azonosítót (SID: Security Identifier), amely egészen a felhasználói fiók megszűnéséig létezik.

Ha a felhasználó hálózati erőforráshoz akar hozzáférni (lásd az alábbi folyamatra UNC kérelem részét), akkor a felhasználó bejelentkezésekor létrejön a SID-et is tartalmazó hozzáférési token a kérelemmel együtt továbbítódik. A fájlrendszer a tokenben lévő SID-et összehasonlítja a célobjektum hozzáférés-engedélyezési listájában (ACL: Access Control List) szereplőkkel. Találat esetén a listában szereplő engedélyeket a felhasználó megkapja, ellenkező esetben azonban egy hibáuzenettel kell béérnie.

A fájlrendszerben beállítható, hogy az egyes mappákhoz és fájlokhoz bizonyos felhasználók, illetve csoportok, milyen jogokkal férhetnek hozzá. Ha adott felhasználónak valamely fájlhoz olvasási és írási jogot adunk, attól más még nem tudja a fájlt írni vagy olvasni.

A fájlrendszer leggyakoribb jogosultsági szintjei (a Windows NT és a Windows 2000 jogosultságmegnevezései kicsit különböznek egymástól, pl. a Change jognak a Modify felel meg Windows 2000-ben):

- **Full Control (teljes hozzáférés):** A felhasználó a fájlokat és a könyvtárakat módosíthatja, törölheti, áthelyezheti és újakat hozhat létre. A fájlok és könyvtárak tulajdonságai és jogosultságai is állíthatók.
- **Change (módosítási jog):** A felhasználó megnézheti és módosíthatja a fájlokat és tulajdonságaikat, ami magába foglalja fájlok létrehozását vagy törlését egy könyvtárban, továbbá fájlok tulajdonságainak módosítását.
- **Read & Execute (olvasási és végrehajtási jog):** A felhasználó a végrehajtható fájlokat és parancsfájlokat futtathatja.
- **List Folder Content (mappalistázási jog):** A felhasználó megnézheti a mappák tartalmát.
- **Read (olvasási jog):** A felhasználó megnézheti a fájlokat és azok tulajdonságait.
- **Write (írási jog):** A felhasználó írhat a fájlokba.
- **No Access (nincs hozzáférés):** Ha egyetlen jelölő sincs kiválasztva, akkor a felhasználó még abban az esetben sem használhatja az erőforrásokat, ha egy magasabb szintű szülőkönyvtárhoz van hozzáférése.

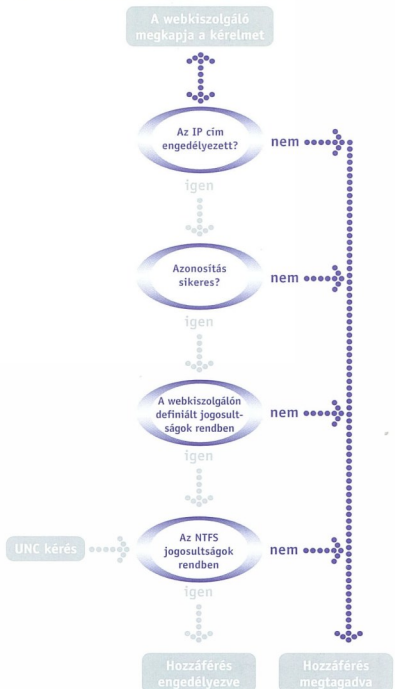
Néhány jogosultságra vonatkozó javaslat, melyek segítenek biztonságos kiszolgálókat készíteni:

- Használjunk NTFS fájlrendszert!
- A rendszergazdának legyen mindenhez teljes hozzáférése, csak nem különleges biztonsági követelmények indokolják ennek az ellenkezőjét.
- A fájlrendszer alapértelmezésben az Everyone (mindenki) csoportot minden új könyvtárhoz automatikusan hozzárendeli teljes hozzáféréssel. Tanácsos ezt azonnal megváltoztatni úgy, hogy megfelelően a biztonsági követelményeknek.

A helytelenül beállított jogosultságok olyan felhasználókat fognak megakadályozni fájlok és könyvtárak elérésében, akiknek erre szükségük van a munkájukhoz. Például hiába van egy felhasználónak olvasási és végrehajtási joga egy programra, ha nincs hozzáférése egy olyan DLL fájlhoz, amit a program használ. Ha a felhasználóknak bizonyos fájlokhoz folyamatosan biztonságos hozzáférést szeretnénk nyújtani, másoljuk ezeket a fájlokat egyazon könyvtárba, majd a könyvtárhoz rendeljük hozzá a megfelelő NTFS jogosultságokat.

### Hogyan működik a web jogosultsági rendszere?

Ha egy kiszolgáló erőforrásait, például fájlokat szeretnénk elérni, akkor a webböngészőnk egy http kérést állít elő, majd elküldi a kiszolgálónak. A kérés tartalmazza például a hitelesítéshez szükséges adatokat, a kérelem eljárás típusát, a kérelem kezdeményezőjének IP címét stb.



Ha a kiszolgáló megkapta a kérést, akkor az IIS először megvizsgálja, hogy a kérelem kezdeményezőjének IP címével lehetséges-e az erőforrás elérése. Ha igen, akkor az IIS ezután rátér a hitelesítési adatok ellenőrzésére. Ha ez is sikeres volt, akkor az IIS azt ellenőrzi, hogy a kérelem eljárás típusa engedélyezett-e az erőforrásra.



A kérés által használt eljárás típusa a HTTP kérelemhez lékellet parancs, amely tudatja az IIS-sel, hogy a felhasználó milyen módon kíván hozzáférni az erőforráshoz. Ha például egy fájl szeretnénk megnézni a kiszolgálón, a böngészőnk ennek megszerzéséhez GET parancsot használna. Ezen a ponton két dolog történhet. Ha az IIS anonymous hitelesítést használ, akkor „imitálja” a felhasználót, és a nevében eljárva próbál meg a kérdéses erőforráshoz hozzáférni. Ha az IUSR\_számitógépnev felhasználói fiókhoz megfelelő jogokat rendelünk, akkor az IIS a kérelmet jóváhagyja, és a fájl megjelenik a felhasználó böngészőjében. Ellenkező esetben az IIS a kérelmet elutasítja, és a böngészőben egy hibaiüzenetet olvashatunk.

Ha a hitelesítés típusa ügyfélbizonyítvány alapú, egyszerű, integrált Windows vagy dígest, akkor az IIS a hitelesítési adatokat átadja a Windowsnak, ami megpróbál a címtárban ennek megfelelő felhasználói fiókot találni.

**Megjegyzés:** *Anonymous HTTP kérelemnél az IIS csak a felhasználónév/jelszó részt nem dolgozza fel, de az IP címre vonatkozó korlátozások, a webes és NTFS jogosultságok továbbra is érvényben maradnak.*

A webkiszolgáló hozzáférési engedélyei beállíthatók minden egyes webhelyre, könyvtárra és fájlra. Ezek minden felhasználóra érvényesek, függetlenül a rájuk vonatkozó egyedi jogosultságoktól. Eszerint, ha egy webhelyre olvasási jogot állítunk be, akkor ez minden felhasználóra vonatkozik (*de az egyes felhasználók NTFS jogosultságai érvényben maradnak*).

#### Webes jogosultsági szintek:

- **Read (olvasási jog):** Alapértelmezett jog, mellyel a felhasználó megnézheti a fájlok tartalmát és a fájlok tulajdonságait.
- **Write (írási jog):** A felhasználó megváltoztathatja a fájlok tartalmát és a fájlok tulajdonságait.
- **Script Source Access (forráskód-hozzáférés):** A felhasználó hozzáférhet a forrásfájlokhoz. Olvasási jog esetén a forráskód olvasható, írásjognál a forráskódba írni is lehet. A forráskód-hozzáférés engedélyezi a fájl forráskódjának, például ASP szkriptek elérését. Csak akkor érvényes, ha az olvasási jog vagy az írási jog engedélyezett.
- **Directory browsing (könyvtárböngészési jog):** A felhasználó megtekintheti a fájlok listáját.

**Figyelem!** *Ha engedélyezzük a forráskód-hozzáférést, akkor a felhasználók az ASP szkriptekből kényes adatokhoz, például felhasználónévhez és jelszóhoz juthatnak hozzá. Rosszabb esetben megváltoztathatják a kiszolgálón futó alkalmazások forráskódját, ami károsan befolyásolhatja a biztonságot és a teljesítményt. Az ehhez hasonló adatok védelme, illetve lehetőségek tiltása leegyszerűbben egyedi felhasználói fiókokkal és magas szintű, például Digest vagy integrált Windows hitelesítéssel oldható meg.*

#### A végrehajtási jogosultságok szintjei:

- **None (semmi):** A kiszolgálón szkriptek, például ASP alkalmazások, és végrehajtható fájlok futtatása nem engedélyezett.
- **Scripts only (csak szkript):** A kiszolgálón csak szkriptek, például ASP alkalmazások futtathatók.
- **Scripts and Executables (szkriptek és végrehajtható fájlok):** A kiszolgálón szkriptek, például ASP alkalmazások, és végrehajtható fájlok egyaránt futtathatók.

#### A WebDAV és a jogosultságok

A HTTP/1.1 protokoll lehetővé tette a felhasználók számára az erőforrások publikálását, zárolását és kezelését a weben. A WebDAV (*Web Distributed Authoring and Versioning*) ezt kibővíti a kérelmek új eljárástípusaival, például a PROPFIND és LOCK parancsokkal. Az új parancskészlettel az ügyfélprogramok képesek az erőforrások és tulajdonságaik módosítására, a fájlok zárolására és a zárolás megszüntetésére, továbbá a fájlok és tulajdonságok szerinti keresésre.

A WebDAV-ot integrálták a Windows 2000-rel és az IIS 5.0-val, ezek biztonsági rendszerét használja, ami magába foglalja az IIS-ben meghatározott jogosultságokat és az NTFS fájlrendszer hozzáférés-engedélyezési listáit (*DACL: Discretionary Access Control List*). Az alábbiakban összefoglaljuk a webes jogosultságokat és ezek hatását a kiszolgálón lévő erőforrások WebDAV segítségével történő hozzáférésére.

- **Olvasás, írás és könyvtárböngészés engedélyezése:** Ezekkel a jogokkal az ügyfél az erőforrásokat listázhatja, módosíthatja (*kivéve azokat, amelyekre nincs írási jog*), saját erőforrásait publikálhatja, és fájlokat manipulálhat.
- **Írás engedélyezése, olvasás és könyvtárböngészés tiltása:** Ha az ügyfelek számára lehetővé akarjuk tenni a magánjellegű adatok publikálását úgy, hogy ezt mások ne láthassák, engedélyezzük az írást, de tiltsuk meg az olvasást és a könyvtárak böngészését. Ez a beállítás különösen akkor hasznos, ha az ügyfelek szavazócédlákat vagy teljesítmény-értékeléseket továbbítanak.
- **Írás és olvasás engedélyezése, könyvtárböngészés tiltása:** Ezt a beállítást akkor használjuk, ha a biztonságot a fájlnevek elrejtésével kívánjuk megvalósítani. Tudnunk kell azonban, hogy ez a biztonságnak egy meglehetősen gyenge szintjét eredményezi, mivel a fájlnevek próbálkozással előbb-utóbb kitalálhatók.
- **Erőforrás-indexelés engedélyezése:** Ne feledkezzünk meg az indexelés engedélyezéséről, ha az ügyfelek számára biztosítani akarjuk a keresés lehetőségét.

#### A webes és NTFS jogosultságok eltérése

Ha a webes jogokat csak olvashatónak állítjuk be, akkor ezzel az NTFS is csak olvasható lesz? Nem. A webes jogok csak azt ellenőrzik, hogy milyen parancsokat használhatunk a HTTP kérésekben. Ha az NTFS jogokat csak olvashatónak állítjuk be, akkor ezzel a webes is csak olvasható lesz? Igen. Ha az NTFS csak olvasható, akkor hiába állítjuk be a webes olvasási és írási jogokat, a HTTP írási kérelem nem sikerülhet.

Tegyük fel, hogy készitünk egy „a” webhelyt, ami egy NTFS kötetben lévő „a” mappára mutat. Beállítjuk a webes írási és



olvasási, illetve az NTF olvasási jogokat. Ekkor hiába próbál egy felhasználó a <http://www.a.com/> címre írni. Ez még akkor sem sikerülne, ha az IIS jóváhagyná az „a” webhelyre, tehát az „a” mappába történő írási kérelmet, mivel az NTF elutasítaná. Ha az „a” webhely hálózatán valaki írni próbál az „a” mappába, az NTF ezt nem engedélyezi. Váltóztassuk meg a jogosultságokat a következőképpen: az NTF írható és olvasható, míg a webes csak olvasható. Ennek eredményeként a webes felhasználó továbbra sem tud írni a webhelyre, mert az IIS megtagadja a kérést. A hálózati ügyfél azonban be tudja illeszteni a fájlt a kérdéses mappába. Az IIS csak a HTTP kéréseket ellenőrzi, a fájlrendszerre vonatkozó kéréseket már nem.

**Alapszabály:** Ha a webes és az NTF jogosultságok eltérőek, akkor a HTTP kéréseknél mindig a szigorúbb érvényesül.

### Jogosultsági varázsló

Hogyan bizonyosodhatunk meg arról, hogy a webes és az NTF jogosultságok tökéletes összhangban vannak? Erre szolgál az IIS jogosultsági varázslója, amely automatikusan összhangba hozza a webes jogosultságokat és az ACL-eket. Használható virtuális könyvtár létesítésére vagy módosítására is.

A varázsló használatakor három dolgot kell szem előtt tartanunk:

- ☞ Jegyezzük fel a virtuális könyvtárak, fájlrendszer-mappák és fájlok biztonsági beállításait, mielőtt megváltoztatjuk őket, mert ez jelentősen megkönnyíti a helyreállítást.
- ☞ A jogosultsági varázsló az érintett fájlok, fájlrendszer-mappák és virtuális könyvtárak webes és NTF jogosultságait is megváltoztatja. Különleges biztonsági követelmények esetén a webes és NTF jogosultságokat a varázsló helyett kézzel kell beállítani.
- ☞ A varázsló kevés választási lehetőséget biztosít a könnyű használat érdekében. Ezek a beállítások jelenleg nem kiterjeszthetők, így nem biztos, hogy minden webhelyre használhatók. Különleges biztonsági követelmények esetén nézzük át a web és fájlrendszer beállításait.

### Jogosultsági problémák és megoldásaik

A webes és az NTF jogosultságok működésének és a közöttük lévő kapcsolat tárgyalása után vessünk egy pillantást a jogosultsági beállításokkal kapcsolatos gyakori problémákra, a valószínű okokra és a megoldásokra. Talán ismerősek lesznek, és remélhetőleg segítenek majd a zavarbaejtőbb problémák elhárításában is.

| probléma  | valószínű ok  | megoldás   |
|---|---|--|
| A weboldal egy része látható, de a képek helyett csak ikonok jelennek meg.      | Lehet, hogy a weboldalhoz tartozó képek egy másik könyvtárban vannak, amelynek jogosultsági beállítása megváltozott.  | Ellenőrizzük a képek könyvtárának NTF jogosultságait és az IIS hitelesítési módszerét!   |
| Az előzőhöz hasonló probléma, de egy bejelentkezési párbeszédablak jelenik meg. | A képek egy másik könyvtárban vannak, amelyre integrált Windows hitelesítést állítottak be. Ez előfordulhat például a FrontPage használatánál, vegyes azonosítású környezetben. A jogosultsági varázsló használata is vezethet ilyen eredményre egy könyvtár kizárásával. | Ellenőrizzük a képek könyvtárának hitelesítési módjait!  |
| A webhelyen publikált anyag nem látható.  | Az azonosítási eszköz feltehetően zárolta a fájlt (LOCK). A feloldása néha 30 másodpercig tart.   | Várjunk egy percig, és próbálkozunk újra!  |
| A weboldalak tartalma WebDAV-val nem változtatható meg.                         | A webes vagy az NTF írási jog nem lett megadva.   | A webes és az NTF jogosultságoknál egyaránt engedélyezni kell az írást és az olvasást. (NTFS-nél a módosítási jog is megfelelő.) |





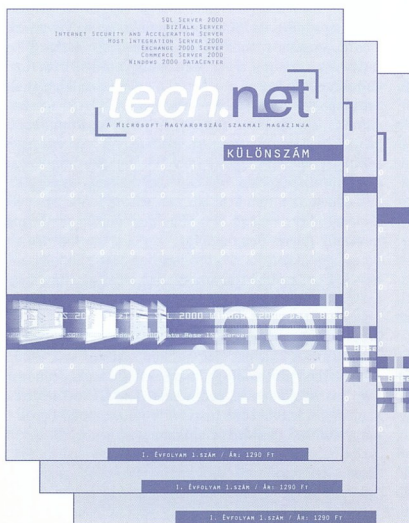
## Összefoglalás

Könnyen kiismerhetjük magunkat a jogosultságok útvesztőjében, ha nem felejtsük el a következőket:

- ☞ Az IIS egy további hozzáférés-ellenőrzési réteget helyez az NTFS fölé, **nem váltja fel az NTFS-t**. Az IIS lehetővé teszi a Windows felhasználói fiókkal nem rendelkező, anonymous felhasználók számára, hogy olyan erőforrásokhoz férjenek hozzá, amelyekhez egyébként nincs engedélyük. Az IIS a Windows fiókkal rendelkező felhasználók számára egy további módot is biztosít az erőforrások elérésére: HTTP felett WebDAV-val.
- ☞ A **webes jogosultságok megváltoztatása nem állítja át automatikusan az NTFS jogokat**, és fordítva, hacsak nem a jogosultsági varázslót használtuk erre a célra. Ha a **webes jogosultságokat kézzel változtattuk meg**, gondoskodnunk kell ezek NTFS megfelelőiről is. Fordítva is igaz: lehet, hogy az NTFS jogok megváltoztatásakor a **webes jogokat is át kell állítani**.
- ☞ Az NTFS jogok csak arra a felhasználóra és csak arra az objektumra vonatkoznak, amelyre beállítottuk. Úgy is fogalmazhatunk, hogy **az egyes felhasználók NTFS jogosultságait objektumonként kell beállítani**. Ezzel szemben a **webes jogok az összes felhasználóra érvényesek**, akik egy adott webhelyet, könyvtárt vagy

fiált szeretnének elérni. **A webes jogok ezért az összes felhasználóra érvényesek objektumonként.**

- ☞ Az NTFS jogosultságok mindenfajta elérésre érvényesek. A webes engedélyek csak a HTTP kérésekre vonatkoznak.
- ☞ Ha UNC kéréssel próbálkozunk, és nincs hozzáférésünk, egy bejelentkezési párbeszédablak jelenik meg, amely engedéllyel rendelkező azonosító megadására szólít fel bennünket. Ha ilyenek nem vagyunk a birtokában, akkor ez az ablak gyakori vendégünk lesz. Megfelelő jogosultság nélküli HTTP kérelemre a „403: Access denied” (hozzáférés megtagadva) hibüzenetet kapjuk.
- ☞ Az IUSR\_számitógépnév fiókot az IIS automatikusan létrehozza anonymous hozzáférési joggal. Ahhoz, hogy „imitálni” tudja a felhasználókat, és biztosítsa számukra az anonymous hozzáférést, az IUSR\_számitógépnév fiókot el kell látnunk helyi bejelentkezési (Log On Locally) joggal, és a Guest csoport tagjává kell tennünk. Ha szeretnénk az IUSR\_számitógépnév fiókot megváltoztatni, például le akarjuk cserélni a jelszavát, akkor készítsünk egy új fiókot, például ANON\_számitógépnév névvel, és használjuk ezt. Az IUSR\_számitógépnév fiók megváltoztatása megjósolhatatlan problémákhoz vezethet.



## Nagy szám a .net különszám!

Még kapható korlátozott számban a .net konferenciára kiadott tech.net különszám, melyben az összes, most megjelent .net Enterprise Server termék újdonságairól olvashat hasznos információkat.

Keressen minket a

<http://technet.netacademia.net> címen!



## Az Active Directory egyszerű kezelése az ADSI segítségével

Valószínűleg már mindenki hallott a Windows 2000 egyik leghasznosabb funkciójáról, az Active Directory-ról. Az Olvasóban felmerülhet a kérdés, hogy miért hasznos ez, miért van szükség rá. Egyszerűen azért, mert igazi Windows 2000 hálózati környezetben a címár használata elkerülhetetlen. Amikor bejelentkezünk egy Windows 2000 tartományba, működésbe lép az Active Directory; ha az irodánk épületében meg akarjuk keresni a legközelebbi színes nyomtatót, a címár segít. A címárobjektumok közötti keresés vagy a 40. épületben dolgozó felhasználók nevének listázása sem jelenthet gondot. Az alkalmazások telepítése is könnyebb: az Active Directory segít az alkalmazások és összetevők (*komponensek*) keresésében és telepítésében. Mindezek mellett ma már smart card-ot is használhatunk a Windows 2000 tartományba történő bejelentkezéshez: a nyílt kulcsú biztonsági infrastruktúra főbb elemei mind-mind az Active Directory-ra támaszkodnak.

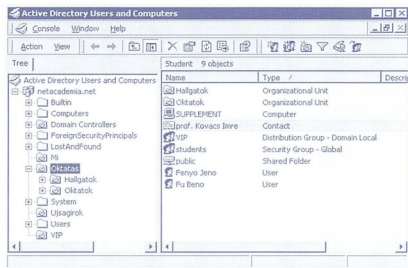
További példánk is vannak: Tetszetős jelentéssel akarjuk elképzelni a főnökünket? Az Active Directory-val ez néhány perc alatt megvalósítható. Összekapcsolhatjuk a Microsoft SQL Server és a címár adatait? Természetesen igen.

A fenti okokon kívül fontos az is, hogy sok más Microsoft technológia, például a Microsoft Message Queue Server (MSMQ), a COM (az objektumosztályok könyvtára, a *Class Store*), az IPsec, a Group Policy Objects (GPO) és az Exchange az Active Directory-t használják. Az Active Directory szolgáltatásait a jövőben egyre több és több alkalmazás fogja kihasználni.

Az Active Directory speciális adatbázis – nem rendszerleíró adatbázis (*mint a registry*) és nem is egy szokásos adatbázis (*mint az SQL Server*) – inkább keresési, kevésbé változtatási és frissítési műveletekre lett optimalizálva. Az Active Directory az adatokat hierarchikusan, többszörözve és bővíthetően tárolja. A többszörözés (*azaz replikáció, több címárkiszolgáló közös működéséhez szükséges szinkronizációs adatfoglalom*) miatt nem tárolhatunk benne dinamikus adatokat, mondjuk Internetes tözsei árfolyamokat, processzor teljesítmény-mutatókat, sem más, hasonló jellegű információt. A számítógép-specifikus adatoknak például ma még inkább a rendszerleíró adatbázisban a helyük. Összességében azt mondhatjuk, hogy a címárban viszonylag statikus, globális és megosztható adatokat tárolunk. A címárban tárolható adatokra tipikus példának a nyomtató-várakozási sorok, a felhasználók adatai és a hálózat, valamint a számítógép beállításai. Az Active Directory címáradatbázis objektumokat tartalmaz (*mégpedig elvileg bármilyet*), minden objektumnak egy vagy több saját jellemzője, attribútuma lehet. Az, hogy a címárban milyen objektumokat tárolhatunk, és azoknak milyen attribútumai lehetnek, a séma (*schema*) határozza meg.

Felmerülhet a kérdés, hogy milyen objektumokat tárol jelenleg az Active Directory. A Windows 2000-ben az Active Directory három partícióból (*más néven névtér, Naming Context*) áll: tartomány (*domain*), séma (*schema*) és konfigurációs (*configuration*) partícióból. A tartománypartíció felhasználói felületre az alábbi ábrán látható. Ez tárolja a felhasználókat, a felhasználói csoportokat, a kapcsolatokat, a számítógépeket, a szervezeti egységeket és sok más objektumtípust. Mivel az Active Directory bővíthető, saját

osztályokat és attribútumokat is adhatunk hozzá. A második partíció, a séma az objektumosztályok és attribútumok meghatározását tartalmazza. Végül a konfigurációs partícióban az Active Directory rendszerszolgáltatások, partíciók és telephelyek beállítási adatait találjuk.



## Az Active Directory tartománypartíciója

### Csatlakozás az Active Directory-hoz

Az Active Directory-hoz több különböző módon is hozzáférhetünk. A Microsoft az Active Directory hozzáférés legfontosabb eszközeként az ADSI használattal ajánlja. Az ADSI a címárakkal való kommunikáció céljára kifejlesztett LDAP protokoll segítségével éri el az Active Directory-t. Kezdjük néhány egyszerű példával. *(Az egyszerűség kedvéért minden példát Microsoft Visual Basic-ben írunk...)* Elsőként csatlakozunk a címárhoz:

```
Set ns = GetObject („LDAP:”)
```

Ennyi az egész. Az ADSI segítségével most (*ha minden jól ment...*) csatlakozunk az alapértelmezett Active Directory tartományvezérlőhöz. Az ADSI az úgynevezett locator szolgáltatás segítségével megkeresi a számunkra legmegfelelőbb tartományvezérlőt (*domain controller, DC*). *(A locator szolgáltatás lényegében a telephelyre vonatkozó információk segítségével határozza meg az adott ügyfél számára legmegfelelőbb tartományvezérlőt.)* Nem kell foglalkoznunk olyan „lánytelen” dolgokkal, mint a kiszolgálónév, hiszen a címárnak éppen az információ központossága lenne az egyik legfőbb célja. Ez a megoldást kiszolgáló nélküli csatlakozásnak nevezzük (*server-less binding*).

De mi van akkor, ha mi történészen szeretjük a kiszolgálóneveket használni? Nos, az ADSI a kiszolgálónév meghatározását is lehetővé teszi.

```
Set obj = GetObject („LDAP://mysrv01”)
```

Más esetben előfordulhat, hogy csak a tartománynevet tudjuk, a kiszolgálónevet pedig nem. Az ADSI a tartománynév megadását is lehetővé teszi. A Windows 2000-ben a tartománynév DNS névként jelenik meg.

```
Set obj = GetObject („LDAP://netacademia.net”)
```

Az ADSI így a netacademia.net tartomány egyik tartományvezérlőjéhez csatlakoztat minket.



## Az ArcadiaBay Corporation

Az ArcadiaBay egy kitalált vállalat (amit kitalált amerikai magyarok vezetnek :-)). Ezt a vállalatot használjuk fel egy szervezetet tipikus helyzeteinek bemutatására.

Az ArcadiaBay nemrég frissítette tartományát Windows NT 4.0-ról Windows 2000-re. Józsi, a rendszergazda, nagyon érdeklődik az Active Directory használatáról. Természetesen ismeri a Windows 2000 Support Tools részeként szállított Active Directory-kezelő eszközöket, de ez a cikk nem az adminisztrációs eszközökről, hanem a programozási hozzáférésről szól. Józsi a telepítés során az „arcadiabay.com” DNS tartománynevet adta meg. A meglévő felhasználók, felhasználói csoportok és számítógépek mind-mind az új arcadiabay.com tartományba kerültek át.

Mielőtt folytatnánk, álljunk meg itt egy kicsit. Először is tudnunk kell, hogy az Active Directory hogyan nevezi el az objektumokat. A fájlneveken keresztül mutatunk egy példát: minden fájl rendelkezik egy névvel és egy elérési útvonallal. A fájlnevek természetesen az adott elérési útvonalon egyedinek kell lenniük. Vegyük a következő példát: „c:\public\specs\adsi25.doc”. Ebben az esetben a fájlnev „adsi25.doc”, a teljes elérési út pedig „c:\public\specs\adsi25.doc.”

Az Active Directory-ban az objektumok ehhez bizonyos szempontból hasonlóan kapnak nevet. Minden objektum rendelkezik egy *objektumnévvel* vagy másképpen *relatív megkülönböztető névvel* (relative distinguished name, RDN), és egy *objektumelérési útvonallal*, azaz *megkülönböztető névvel* (distinguished name, DN). Az RDN vagy objektumnév két részből áll: az attribútumazonosítóból (attribute ID) és magából az értékből. Például DC=ArcadiaBay. A DC egy RDN attribútumazonosító, amely a tartomány összetevőre utal, az ArcadiaBay pedig az attribútum értéke. Esetünkben az Arcadiabay tartományobjektum megkülönböztető neve DC=ArcadiaBay, DC=Com. Látható, hogy a DN több, egymástól elválasztott RDN-ből áll, amelyek mindegyike azonosítót (például DC) és értéket (például ArcadiaBay) is tartalmaz.

Ezek szerint a tartományobjektumhoz a következőképpen kapcsolódhatunk:

```
Set dom =
GetObject („LDAP://DC=ArcadiaBay,DC=Com“)
```

A tartományobjektum megtalálása után kíráthatjuk annak néhány attribútumát:

```
Debug.Print dom.Get („Name“)
Debug.Print dom.Get („whenCreated“)
```

## Szervezeti egység (Organizational Unit, OU) létrehozása

Maradjunk a tartományobjektumnál, tegyük egy kicsit érdekesebbé a helyzetet. Az ArcadiaBay Értékesítési (Sales) és Termelési (Production) osztályból áll. A vállalat azt tervezi, hogy felvesz két Windows 2000 rendszergazdát, mindkét osztályra egyet. Józsi, a vállalat fő rendszergazdája, két új szervezeti egységet hoz létre az ArcadiaBay tartományban. A szervezeti egység létrehozásával Józsi sok objektumot csoportosíthat, és lehetővé teheti, hogy azokat valaki más kezelje („kiadja” a munkát). A Sales szervezeti egységet (Organizational Unit, OU) a következőképpen hozhatjuk létre:

```
Set salesOrg =
dom.Create („organizationalUnit“,
„OU=Sales“)
```

```
salesOrg.Put „description“, „Sales
Headquarter, SF“
```

```
salesOrg.Put „wwwHomePage“,
„http://arcadiabay.com/sales“
salesOrg.SetInfo
```

Jöjjen egy kis magyarázat. A Create eljárás adja meg az osztálynevet és az új objektum nevét. A visszakapott objektum attribútumai a Put metódus hívásával módosíthatók, ehhez meg kell adnunk a módosítandó attribútum nevét és értékét. Az objektum és a benne történt módosítások egyelőre nem a cím táblaiban jöttek létre, hanem a memóriában, egy gyorsított tábla. A SetInfo eljárás meghívásakor a változtatások (ebben az esetben az objektum létrehozása és az attribútum módosítása) bevésszödnék a cím tábla. A változtatás tranzakcióalapú, ami azt jelenti, hogy ha az adatok mentése során valami hiba történne (elmenne az áram), a rendszer biztos, hogy nem kerül „félállapotba”. Esetünkben ez azt jelenti, hogy vagy létrejön az új objektum az összes változtatással együtt, vagy nem változik semmi (ekkor az objektumot sem fogjuk találni). Belátható, hogy a sok rossz közül ez a legkevésbé kellemetlen eset.

A Termelési osztály szervezeti egység létrehozása legyen így feladat. Egyéb más illeszthetők a szervezeti egységek? Igen! Tétélezhetük fel, hogy a Sales osztályt tovább bontjuk a keleti (east) és nyugati (west) régió szerint.

```
Set east =
salesOrg.Create („organizationalUnit“,
„OU=East“)
east.SetInfo
```

Ugyanezt a nyugati régióval is megtehetjük.

Ha közvetlenül a Sales osztály keleti régiójához akarunk csatlakozni, adjuk meg a szervezeti egység teljes megkülönböztető nevét.

```
Set east = GetObject („LDAP://OU=East,
OU=Sales, DC=ArcadiaBay,DC=COM“)
Debug.Print east.Get „description“
east.Put „wwwHomePage“,
„http://arcadiabay.com/sales/east“
```

Ha már csatlakoztunk a szülőobjektumhoz (Sales), a szülőobjektumból is csatlakozhatunk a származtatott objektumhoz (East).

```
Set east =
salesOU.GetObject („organizationalUnit“,
„OU=East“)
```

Honnan tudjuk, hogy tényleg létrehoztuk-e ezeket az új objektumokat? Használjuk az Active Directory Users and Computers MMC modult és megláthatjuk. Az új szervezeti egységeknek ott kell lenniük.

**Meglévő felhasználók telepítése a szervezeti egységbe**  
 A Windows NT 4.0 tartomány frissítésekor az összes felhasználó és csoport az ArcadiaBay tartomány „Users” mappájába került. Helyezzünk át néhány felhasználót a számukra megfelelő szervezeti egységbe. Az ADSI használatával egyébként egy objektumot más Windows 2000 tartományokba is áthelyezhetünk.

```
Set usr =
salesOU.MoveHere („LDAP://CN=mikeg,CN=Users,
DC=ArcadiaBay,DC=COM”, vbNullString)
```

A **MoveHere** megkapja az áthelyezendő objektum **AdsPath**-jét (a szolgáltató nevét, amely a következőkből áll: *LDAP és egy megkülönböztető név; ez valójában az ADSI objektum egyedi neve, ezzel az attribútummal minden ADSI objektum rendelkezik*) és az új objektumnevet (*RDN*). Ha az eredeti nevet akarjuk megtartani, egy nullát adjunk meg. Ha az objektumot át akarjuk nevezni, az új nevet a második paraméterként adhatjuk meg. A példában Michael Grayt (*mikeg*) helyezzük át a Sales szervezeti egységbe.

Az **AdsPath** a részletes leírása megtalálható az MSDN Library-ban, az ADSI dokumentációban.

#### Új felhasználók létrehozása a szervezeti egységben

A Sales szervezeti egységbe felvettek egy új, Julie Adam nevű alkalmazottat. Ő lesz Michael Gray új főnöke. Jözsini (ahelyett, hogy két kattintással megtenné :) ) billentyűzetet ragad, és gépelni kezd:

```
Set salesOU =
GetObject („LDAP://OU=Sales,DC=ArcadiaBay,
DC=COM”)
Set usr = salesOU.Create („user”,
„CN=Julie Adam”)
usr.Put „samAccountName”, „juliea”
usr.Put „userPrincipalName”,
„julie@arcadiabay.com”
usr.Put „title”, „Marketing Manager”
usr.SetInfo

usr.SetPassword „seahorse”
usr.AccountDisabled = False
usr.SetInfo
```

Meg kell adnunk tehát például a **samAccountName** attribútumot – ez a felhasználói osztály egyik kötelező attribútuma, az objektum létrehozása előtt pedig minden kötelező attribútumot ki kell töltenünk. A felhasználó **samAccountName** attribútuma a Windows NT 4.0-es Windows 9.x számítógépekről történő bejelentkezéskor használt felhasználónevet tartalmazza. Természetesen ezt a nevet a Windows 2000-ben is használhatjuk. Mi vajon a **userPrincipalName**? A Windows 2000 környezetben (ha mind az ügyfél, mind a tartományvezérlő Windows 2000-et futtat), az úgynevezett **User Principal Name**-mel (**UPN**) is bejelentkezhetünk. Példánkban Julie UPN-je julie@arcadiabay.com.

Jözsi jogaival élve a **SetPassword** eljárással a jelszót is megadhatja. A **SetPassword** eljárás csak akkor működik, ha az objektumot a cím tárban már ténylegesen létrehoztuk (emlékezzünk a gyorsítótárra!). Ezért mielőtt beállítjuk a felhasználó jelszavát, meg kell hívunk a **SetInfo** metódust. Végül az **AccountDisabled** tulajdonság hamisra

(*FALSE*) állításával engedélyezzük a felhasználó fiókját. Tegyük Julie-t Michael főnökévé.

```
Set usr =
GetObject („LDAP://CN=mikeg, OU=Sales,
DC=ArcadiaBay, DC=COM”)
usr.Put „manager”, „CN=Julie
Adam,OU=Sales, DC=ArcadiaBay,DC=COM”
usr.SetInfo
```

Itt felmerülhet egy lehetséges probléma. Mi történik akkor, ha Julie megváltoztatja a nevét, átkerül egy másik szervezeti egységbe vagy kilép a vállalatból? Ki tartja karban ezt a közvetlen beosztotti kapcsolatot? Ezt majd később, az átszervezési gyakorlatnál tekintjük át. Még egy megjegyzés: mivel az Active Directory séma bővíthető, az objektumok modellezésénél könnyen létrehozhatunk hasonló közvetlen beosztotti kapcsolatokat.

A következő feladat előtt nézzük meg, hogy kik Julie közvetlen beosztottai.

```
Set usr =
GetObject („LDAP://CN=Julie Adam,
OU=Sales, DC=ArcadiaBay, DC=COM”)
reports = usr.GetEx („directReports”)
```

```
For Each directReport in reports
Debug.Print directReport
Next
```

Azt látjuk, hogy a közvetlen beosztott Michael G, pedig nem is állítottuk be a **directReports** attribútumot. Az Active Directory „automagikusan” kezeli a főnök-beosztott viszonyt. Mi az a **GetEx** eljárás? A cím tárak világában egy attribútumnak egy vagy több értéke lehet. Mivel tudjuk, hogy a **directReports** többértékű (ez a sémából és a dokumentációból is látszik), egyszerűbb a **GetEx** használata, amely mindig egy eredménysort ad vissza, függetlenül attól, hogy egy vagy több értékről van-e szó. A For Each segítségével pedig egyenként listázhajuk az eredményeket. A közvetlen beosztotti kapcsolatok az Active Directory Users and Computers MMC modulban, a felhasználó tulajdonságlapján is megtekinthetők.

#### Új csoport létrehozása

Jözsinek egy új csoportot kell létrehoznia. A csoport tagjai számára azután elérhetővé akar tenni néhány erőforrást (fájlokat, Active Directory-beli vagy más objektumokat).

```
Set ou =
GetObject („LDAP://OU=Sales,DC=ArcadiaBay,
DC=COM”)
Set grp = ou.Create („group”,
„CN=Management”)
grp.Put „samAccountName”, „mgmt”
grp.Put „groupType”,
ADS_GROUP_TYPE_DOMAIN_LOCAL_GROUP Or
ADS_GROUP_TYPE_SECURITY_ENABLED
grp.SetInfo
```

Ezt a „Management” nevű csoportot a Sales szervezeti egységben hoztuk létre. Először is a Sales szervezeti egység ADSI objektumára van szükségünk. A **samAccountName** a kompatibilitáshoz szükséges kötelező



attribútum. Példánkban a Windows NT 4.0-s eszközök (például a Felhasználókezelő) számára a csoport neve „mgmt”, és nem „Management”. A csoport típusát is meg kell határoznunk. A Windows 2000 tartományban három csoporttípust használhatunk: globális (Global), tartományi helyi (Domain Local) és univerzális (Universal) csoportot, emellett a csoporthoz adatvédelmi jellemzők is tartoznak. A csoport lehet biztonsági (Security) vagy adatvédelem nélküli. A biztonsági csoportok az erőforrások hozzáférési jogainak listájában szerepelhetnek (csakúgy, mint az egyszerű felhasználók). A másik fajta, úgynevezett terjesztési (vagy disztribúciós) listák nem használhatók ugyanígy – egy terjesztési listának például nem adhatunk hozzáférési jogokat az erőforrásokhoz. A rendszer frissítése során a Windows NT 4.0-ból származó csoportok biztonsági csoportokká alakultak. Az Active Directory adatvédelem nélküli csoportjai az Exchange terjesztési (levelezési) listáihoz hasonlítanak, a Windows 2000-ben nagyon hasonló a csoportok és a terjesztési listák létrehozásának művelete. A Windows 2000 címítár natív üzemmódjában a csoportok bármilyen szintig egymásba illeszthetők. Ha a címítár vegyes (mixed) módban üzemel, a tartományvezérlők között még szerepelhet Windows NT 4.0 kiszolgáló is.

#### Felhasználók hozzáadása egy csoporthoz

Adjuk hozzá Julie-t a Management csoporthoz.

```
Set grp =
GetObject („LDAP://CN=Management,
OU=Sales, DC=ArcadiaBay, DC=COM“)
grp.Add („LDAP://CN=Julie Adam,
OU=Sales, DC=ArcadiaBay, DC=COM“)
```

A fenti példa magáért beszél: Add eljárásnak átadtuk Julie azonosítóját, így ő bekerült a csoportba.

#### Az objektumok számlálása

Gyakran van szükség arra, hogy megszámloljuk a címítár egy mappájában vagy szervezeti egységében tárolt objektumokat. A mappa gyermekobjektumainak számbavétele során kilistázzuk a címítár-hierarchiában közvetlenül a tároló alatt található objektumokat. A fájltrendszerben ez annak felel meg, amikor egy adott könyvtár fájljait ellenőrizzük. Ha akarunk, egy szinttel feljebb is léphetünk, így egy adott objektum szülőobjektumát tekinthetjük meg.

```
Set ou = GetObject („LDAP://OU=Sales,
DC=ArcadiaBay, DC=COM“)
For Each child in ou
    Debug.Print child.Name
Next
```

Az objektumok számlálása során kiszűrhetünk adott típusú objektumokat. Ha például csak a felhasználókat és a csoportokat akarjuk megtekinteni, a listázás előtt (For Each) állítsuk be a szervezeti egység Filter attribútumát, aminek egy osztálynevekből álló fűzért kell átadnunk (erre való az Array()):

```
Ou.Filter = Array („user“, „group“)
```

Egy objektum szülőobjektumának nevét a Parent attribútum tartalmazza. Ha a név már megvan, könnyen csatlakozhatunk a szülőobjektumhoz is:

```
parentPath = obj.Parent
Set parent = GetObject(parentPath)
```

#### Objektumok keresése

Julie-nak meg kell keresnie a 101. osztályon dolgozó összes programmenedzser telefonszámát. Segítsünk neki létrehozni erre egy olyan script-et, amely az adatbázisok eléréséhez fejlesztett programozói felületet, az ADO-t (ActiveX Data Objects) és az ADSI-t is használja.

```
'ADO Connection és Command objektum létrehozása
Set con = CreateObject („ADODB.Connection“)
Set com = CreateObject („ADODB.Command“)
' A kapcsolat megnyitása
con.Provider = „ADSsOObj“ 'Ez az
ADSI-OLEDB provider neve
con.Open „Active Directory Provider“
' A Command objektumot hozzárendeljük a
létrejött kapcsolathoz
Set Com.ActiveConnection = con

'A keres?kérés összeállítás
Com.CommandText = „SELECT
name,telephoneNumber FROM
'LDAP://DC=ArcadiaBay, DC=com' WHERE
objectCategory='Person'
AND objectClass = 'user'
AND title='Program Manager'
AND department=101“
```

```
'A keresés futtatása
Set rs = Com.Execute
' Az eredménytábla (RecordSet) listázása
While Not rs.EOF
    Debug.Print rs.Fields(„Name“) & „ „ &
rs.Fields(„telephoneNumber“)
    rs.MoveNext
Wend
```

Huh, ebben a cikkben ez az első tíz sornál hosszabb programkód. Ha az olvasó ismeri a Microsoft ActiveX Data Objects-et (ADO), ez gyerekeitének tűnhet. Visual Basicben vagy scriptkörnyezetben egy ADSI keresés végrehajtásához három dologra van szükség: kapcsolatra (Connection), parancsra (Command) és eredménytáblára (Recordset). A Connection objektumban megadhatjuk a címítárkiszolgáló nevét és más jellemzőket. A Command objektumban a keresési feltételeket és a lekérdezés szövegét adhatjuk meg, gyakorlatilag magát a parancsot. A lekérdezés végrehajtása előtt a Connection objektumot össze kell kapcsolnunk a Command objektummal. Végül a Recordset objektum használható az eredményhalmaz kezelésére.

Az ADSI a lekérdezés szövegezésének két formáját támogatja. Az előbbi példa az SQL formát mutatja be, azonban az LDAP formát is használhatjuk. Az LDAP lekérdezési szö

veg az RFC 2254 szabványra épül. Az előző példa a következőképpen fordítható le LDAP lekérésére:

```
Com.CommandText =
"<LDAP://DC=arcadiaBay,DC=COM>; (&(objectCategory=Person)(objectClass=user)(title=ProgramManager)(department=101);name,telephoneNumber;subTree"
```

Mit jelent a subTree szó a szöveg végén? A címár világában megadhatjuk a keresés hatókörét. A következőkből választhatunk: *base (alap)*, *oneLevel (egyszintű)* és *subTree (alkönyvtáras)*. A base csak magát az objektumot olvassa be; a oneLevel a „dir” parancshoz hasonlóan a közvetlen származtatott objektumokat is; a subTree pedig a dir /s-hoz hasonlóan több szinten is keres.

Az SQL forma használatával a hatókör a parancs tulajdonságai között adható meg. Például: Command.Properties („Search Scope”) = ADS\_SCOPE\_ONELEVEL. Ha nem adjuk meg a hatókört, az alapértelmezésben subTree keresés lesz.

### Átszervezés

Az egész Sales szervezeti egység átkerül egy új szervezeti egységbe – a „Sales and Support”-ba. Julie-t *(nyilván az általa írt lekérdezőscript hatására)* előléptetik alelnökké, ő lesz az új szervezeti egység vezetője.

```
Set dom = GetObject („LDAP://DC=ArcadiaBay,DC=COM")
Set salesSupport =
dom.Create („organizationalUnit", „CN=Sales and Support")
Set sales =
salesSupport.MoveHere („LDAP://OU=Sales,DC=ArcadiaBay, DC=COM", vbNullString)
```

Három sor kód egy átszervezéshez nem rossz, ugye? Így a sales szervezeti egység minden objektuma, a szervezeti al-egységeket is beleértve, átkerült az új szervezeti egységbe. Helyezzük át Julie-t a Sales and Support szervezeti egységbe.

```
Set usr =
salesSupport.MoveHere („LDAP://CN=Julie Adam,OU=Sales, OU=Sales and Support,DC=ArcadiaBay,DC=COM")
usr.Put „title", „Vice President"
usr.SetInfo
```

Felmerülhet a kérdés, hogy mi történt a Julie és Michael közötti közvetlen beosztotti kapcsolattal. „Ki teszi rendbe ezt a kapcsolatot?” – kérdezhetjük. Mint azt talán az Olvasó már ki is találta, az Active Directory „automágikusan” elintézte ezt.

### Heterogén adatbázisok adatainak összekapcsolása

A szervezetek adataikat általában több heterogén adatbázisban tárolják: a humán erőforrások adatait például SQL Server-en, a felhasználói fiókokkal kapcsolatos információkat a címárban, egyéb információkat pedig más, egyedi formátumokban. Így az adatok eléréséhez szükséges különböző programozói felületek, protokollok és adatformátumok használata miatt a több forrásból származó adatok egyetlen használható jelentésbe való összevonása nem egyszerű. Az SQL Server 7.0 Distributed Query-vel, az ADSI-val, az

OLEDB-vel és az Active Directory-val lehetővé válik az Active Directory és az SQL Server adatainak összekapcsolása. Az összevont adatokat később meg is tekinthetjük. Ennek lépése:

### Az SQL Server-ben

1. Futtassuk a Query Analyzer-t (*Start | Programs | Microsoft SQL Server 7.0*)
2. Jelentkezzünk be az SQL Server számítógépre.
3. Hajtsuk végre a következő sort (*jelöljük ki és nyomjunk CTRL+E-t*):

```
sp_addlinkedserver 'ADSI', 'Active Directory Service Interfaces', 'ADSDSOObject', 'adsdatasource' go
```

Ez utasítja az SQL Server-t, hogy az „ADSI” szót kapcsolja össze az ADSI OLE DB szolgáltatóval, az „ADSDSOObject”-tel. Így az SQL Server-ből hozzáférhetünk az Active Directory-hoz.

4. Írjuk be és hajtsuk végre a következőket:

```
SELECT * FROM OpenQuery( ADSI, 'SELECT name, adsPath FROM „LDAP://DC=ArcadiaBay,DC=com" WHERE objectCategory = „Person" AND objectClass= „user",,)
```

5. Az ADSI LDAP formát is használhatjuk. Példa:

```
SELECT * FROM
OpenQuery(ADSI, '<LDAP://DC=ArcadiaBay;DC=COM>; (&(objectCategory=Person)(objectClass=user));name, adsPath;subtree')
```

### Nézet létrehozása és megtekintése

Nézetet is hozhatunk létre az Active Directory-ból származó adatok megtekintésére. Fontos, hogy csak a nézet definíciója tárolódik az SQL Server-en és nem a tényleges eredményhalmaz. Így előfordulhat, hogy egy-egy későbbi végrehajtás során más eredményt kapunk. A nézet létrehozásához a következőket írjuk be és hajtsuk végre:

```
CREATE VIEW viewADUsers AS
SELECT * FROM OpenQuery(
ADSI, '<LDAP://DC=ArcadiaBay,DC=com>; (&(objectCategory=Person)(objectClass=user));
name, adsPath,title;subtree')
```

A nézet megtekintéséhez a következőket írjuk be:

```
SELECT * from viewADUsers
```



## Heterogén kapcsolat az SQL Server és az Active Directory között

Az ArcadiaBay alkalmazottait hathavonta értékelik. Az értékelés eredményét az SQL Server humán erőforrás (*Human Resource*) adatbázisában tárolják. Először is hozunk létre egy táblázatot az alkalmazottak teljesítményének áttekintéséről:

```
CREATE TABLE EMP_REVIEW
(
  userName varchar(40),
  reviewDate datetime,
  rating decimal
)
```

Írjunk be néhány bejegyzést:

```
INSERT EMP_REVIEW VALUES('Julie Adam',
'2/15/1999', 4 )
INSERT EMP_REVIEW VALUES('Julie Adam',
'7/15/1999', 5 )
INSERT EMP_REVIEW VALUES('Michael Gray',
'2/15/1999', 3 )
INSERT EMP_REVIEW VALUES('Michael Gray',
'7/15/1999', 4 )
```

Kapcsoljuk össze a kettőt – az Active Directory felhasználói objektumait és az SQL Server táblázatot:

```
SELECT AdsPath, userName, title,
ReviewDate, Rating
FROM EMP_REVIEW, viewADUsers
WHERE userName = Name
```

Voilà! Mind az SQL Server-ből, mind az Active Directory-ből megkaptuk az eredményeket. Az AdsPath és a Title oszlop az Active Directory-ből, míg a userName, a ReviewDate és a Rating az SQL táblázatból származik. Ehhez a kapcsolat-hoz más nézetet is létrehozhatunk:

```
CREATE VIEW reviewReport
SELECT AdsPath, userName, title
ReviewDate, Rating
FROM EMP_REVIEW, viewADUsers
WHERE userName = Name
```

## RootDSE

Minden címtárkiszolgáló rendelkezik egy **RootDSE** nevű egyedi bejegyzéssel. Ez a kiszolgálóról szolgáltat információkat – például a kiszolgáló szolgáltatásairól, az általa támogatott LDAP verzióról, a rajta található névtérről stb. Tétélezük fel, hogy egy olyan scriptet vagy alkalmazást szeretnénk létrehozni, amely bármilyen Windows 2000-tartománykörnyezetben képes futni. Az Active Directory-hoz történő csatlakozáskor a megkülönböztető nevet, a kiszolgálónevet vagy a tartománynevet is megadhatjuk. Mi van akkor, ha nem rendelkezünk ezekkel? Ekkor a **RootDSE** objektum siet a segítségünkre. A következő példa bármilyen tartományban megváltoztatja a tartományleírást:

```
Set rootDSE =
GetObject(„LDAP://RootDSE”)
Set dom = GetObject(„LDAP:/// &
rootDSE.Get(„defaultNamingContext”)
)
dom.Put „description”, „My domain”
dom.SetInfo
```

A **defaultNamingContext** attribútummal a **RootDSE**-ben az aktuális tartományhoz csatlakozhatunk (az *Arcadia Bay*-ben a **defaultNamingContext: dc=ArcadiaBay, DC=COM**).

## Szervezeti egységek felügyeletének delegálása

Az ArcadiaBay két rendszergazdát (*Miki és Pali*) vett fel az East és a West szervezeti egység kezelésére. Jözsi lehetővé teszi számukra, hogy saját szervezeti egységükben ők hozhassanak létre és törölhessenek le felhasználókat:

```
Set ou = GetObject(„LDAP://OU=East,
OU=Sales, DC=ArcadiaBay,DC=COM”)
Set sec = ou.Get(„ntSecurityDescriptor”)
Set acl = sec.DiscretionaryAcl
```

```
'vagy: Set ace = new
ADsAccessControlEntry
Set ace =
CreateObject(„AccessControlEntry”)
```

```
'Engedélyezés
ace.AceType =
ADS_ACETYPE_ACCESS_ALLOWED_OBJECT
'Létrehozás és törlés jog
ace.AccessMask =
ADS_RIGHT_DS_CREATE_CHILD Or
ADS_RIGHT_DS_DELETE_CHILD
'A felhasználó objektumostály GUID-ja a
sémában:
ace.ObjectType = „{BF967ABA-ODE6-11D0-
A285-00AA003049E2}”
ace.AceFlags = ADS_ACEFLAG_INHERIT_ACE
'Jog öröklésének beállítása
ace.Flags = ADS_FLAG_OBJECT_TYPE_PRESENT
ace.Trustee = „ARCADIABAY\Miki” '? a
kedvezményezett
acl.AddAce ace
```

```
sec.DiscretionaryAcl = acl
ou.Put „ntSecurityDescriptor”, Array(sec)
ou.SetInfo 'Tárolás az Active Directoryban
```

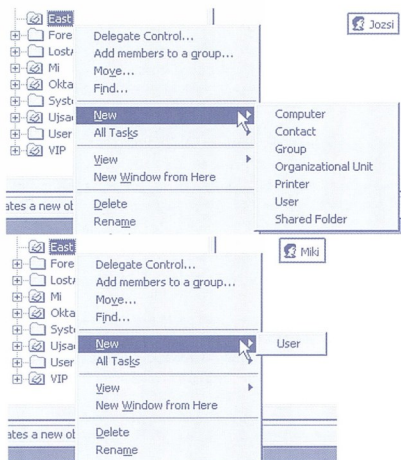
```
Set ace = Nothing
Set acl = Nothing
Set sec = Nothing
```

Egy kicsit komplikált, ugye?

Az Active Directory-ban (*egyébként a Windows NT-ben és Windows 2000-ben is – a szerk.*) minden objektum rendelkezik egy úgynevezett adatvédelmi leíróval (*Security Descriptor*). Az adatvédelmi leíró határozza meg, hogy az objektummal ki és mit művelhet. Maga az adatvédelmi leíró két hozzáférési listát (*Access Control List, ACL*) tartalmaz, amelyeket *Discretionary ACL*-nek (*DACL*) és *System ACL*-nek (*SACL*) nevezünk. Minden ACL hozzáférési bejegyzéseket (*Access Control Entry, ACE*) tartalmazhat. Maga az

ACE határozza meg egy objektumhoz való hozzáférési jogukat. Az ACE a jogosult személye mellett a számára engedélyezett vagy éppen tiltott műveletet is meghatározza. Ilyen műveletek például a gyermekobjektum törlése vagy létrehozása, az objektumjellemzők olvasása vagy írása. Azt is meghatározhatjuk, hogy az ACE milyen objektumosztályokra vagy attribútumokra fejtsse ki a hatását. Példánkban a *user* (felhasználó) osztályt választottuk. Ezután a következő kérdésre kell válaszolnunk: „Kire vonatkozik ez az ACE?” Példánkban Mikit adtuk meg. Végül meg kell határoznunk az ACE öröklődési tulajdonságait – megadható például, hogy az ACE a hierarchiában öröklődjön.

Összességében: az előbbi példa azt eredményezi, hogy Miki képessé válik felhasználói objektumokat létrehozni és törölni az East Sales szervezeti egységben. Az alábbi ábra az Active Directory **Create New View** menüjét mutatja a kód futtatása után. Ha Józsiék (*rendszergazdaként*) jelenkezünk be, több létrehozható osztályt is láthatunk. Ha Mikiként, csak felhasználói objektumokat hozhatunk létre.



Az új objektum létrehozására szolgáló menü két különböző felhasználó számára

### A globális katalógus (Global Catalog)

Több tartományt összekapcsolhatunk úgy, hogy tartományi fát (*tree*) alkossanak, több tartományfából pedig tartományi erdőt (*forest*) hozhatunk létre. A közös erdőben található particiók ugyanazt a séma- és beállítási particiót használják. Az erdő minden objektumáról egy globális katalógusban (*Global Catalog, GC*) lista készül, de egy adott erdőhöz egy-nél több globális katalógus is tartozhat. A globális katalógus mindig tartományvezérlőn található, de egy tartományvezérlő nem feltétlenül üzemeltet globális katalógust. A globális katalógusba *nem minden attribútum kerül be*, csak az általunk kiválasztott attribútumok jelennek meg benne.

A globális katalógus elsődleges célja a vállalati szintű keresések biztosítása: ha például Józsi szeretné tudni, kik azok a felhasználók, akiket az elmúlt 30 napban vettek fel. Ha programból szeretnénk kapcsolódni a globális kataló-

gushoz, az LDAP: helyett a GC-t kell használnunk:

```
Set gc = GetObject(„GC://srv01.arcadiabay.com“)
Set gc = GetObject(„GC://DC=Arcadiabay, DC=COM“)
```

Ha az Europe.ArcadiaBay.com az ArcadiaBay.com egy gyermektartománya, a **GC://DC=ArcadiaBay,DC=Com** globális katalógus a Europe.arcadiabay.com objektumait is magába foglalja.

### Az ADSI bővítmények

Az ADSI bővítmények segítségével a címár egy objektum-osztályát összekapcsolhatjuk saját COM objektumunkkal. Aki az objektumot ezután ADSI-n keresztül éri el, úgy érzékeli, mintha az adott attribútum vagy metódus az ADSI objektum sajátja volna.

Például: amikor az ArcadiaBay-hez egy új alkalmazott lép be, a rendszergazda a címárban egy új felhasználói objektumot hoz létre, a bérelszámoló pedig bejegyzést készít a humán erőforrás adatbázisban. Az ADSI bővítménnyel ez a folyamat egyetlen scriptbe foglalható.

```
Set usr = ou.Create(„user“, „CN=Alice Johnson“)
// ... attribútumok beállítása ...
usr.SetInfo
```

`usr.AddToPayroll` ' ez egy ADSI bővítmény-metódus

`Debug.Print „A(z) „ & usr.Name & „ felhasználót létrehoztam“`

### További olvasmányok:

Active Directory Services Interfaces 2.5 telepítő (ADSI 2.5 és providerek)

W2K: ADSI 2.5 telepítésére nincs szükség

NT4: <http://www.microsoft.com/ntserver/nts/downloads/other/ADSI25/x86.asp>

W98: <http://www.microsoft.com/ntserver/nts/downloads/other/ADSI25/w98.asp>

W95: <http://www.microsoft.com/ntserver/nts/downloads/other/ADSI25/w95.asp>

MSDN Library: <http://msdn.microsoft.com/library>



## Microsoft

E hónapban is – már-már hagyományyszerűen – a NetAcademia levelezési listák elmúlt havi forgalmából meszólunk. Hisszük, hogy az itt felmerülő kérdések nagy része szélesebb körű nyilvánosságot is érdekelhet. Ha Önnek is kérdése van, és kíváncsi az ország többszáz szakértőjének a véleményére, iratkozzon fel levelezési listánkra a (<http://www.netacademia.net/?page=lyris.htm>) címen, és kérdezzen bátran!

**K:** Megjelent a Windows 2000 Service Pack 1. Az előzetes leírásokban volt arról szó, hogy a Windows 2000 javítócsomagjait be lehet építeni a telepítőkészletbe, hogy később már ne kelljen ezzel is kinlódni. Hogy is van ez?

**V:** A Service Pack 1 képes beépülni a Windows 2000 eredeti telepítőkészletébe. A kulcs az update.exe kapcsolóiban rejlik. A -z kapcsoló hatására például elmarad a telepítés utáni újraindítás, a -q esetén felhasználói beavatkozás nélkül, csendben folyik a telepítés, a -n megadásakor pedig nem készül biztonsági másolat az eredeti fájlokról (*helytakarekosság...*). Végül itt van a -s: mappanév kapcsoló, ahol megadhatjuk az eredeti telepítőkészlet i386 könyvtárát tartalmazó mappájának elérési útját. (*Tehát, ha a telepítőkészlet az E:\W2KSETUP\i386 könyvtárban található, akkor a parancs így néz ki: update.exe -s:E\W2KSETUP*).

**K:** Ha nekem a hálózatos SP1 telepítőkészlet van meg (*sp1network.exe*), akkor mit tehetek?

**V:** Az sp1network.exe a -x kapcsoló segítségével kitömöríthető úgy, hogy maga a telepítés nem kezdődik meg (*csak létrejön a javítócsomag telepítőjét tartalmazó csomag, benne az update.exe-vel*). Ha az sp1network.exe-nek bármilyen más kapcsolót megadunk, az továbbbővíti az update.exe felé (*magyarán szólva, az első válaszban az update.exe helyére bárhova nyugodtan behelyettesíthetjük az sp1network.exe-t is*).

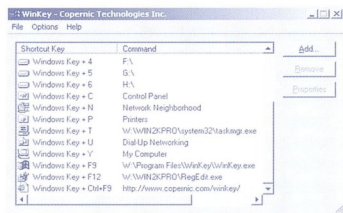
**K:** Van valami érdekesség a javítócsomagok között megjelent gyorsjavításokkal is? Engem például a hideg kiráz, amikor arra gondolok, hogy a tizenkét hotfix telepítésekor tizenkétyszer újra kell indítani a számítógépet...

**V:** Természetesen, hiszen különben a kérdés nem szerepelne ebben a rovatban :-). A hotfix.exe mindig más néven je-lentkezik: Q#####\_XXX\_YYY\_ZZZ\_LL.exe, ahol ##### a hiba leírását tartalmazó tudásbázis-cikk száma (*ld. Microsoft Knowledge Base*), XXX az operációs rendszer jele, YYY a javítócsomag, amiben (*majd*) megtalálható a javítás, a ZZZ a hardverplatform, LL pedig a nyelv jele. Engedtessek meg, hogy a továbbiakban egyszerűen hotfix.exe-ként hivatkozzunk rá.

A hotfix.exe kapcsolói többek között: -y: uninstall (*ügyeljünk a megfelelő sorrendre!*), -n: nincs biztonsági mentés, -z: nincs újraindítás, -t: a telepített javítások listája (!). Ez utóbbi kapcsolónak egyébként könnyű dolga van, hiszen minden telepített hotfix a HKLM \Software\Microsoft\Windows NT\Current Version\Hotfix-<#####> címszó alatt bejegyzi magát a registrybe.

**K:** Van néhány kellemes funkció a felhasználói felületben, amit a (Windows) gomb segítségével könnyen elérhetünk, például a W+M (*minden ablak minimalizálása*), vagy a W+E (*explorer indítása*). Van arra valami mód, hogy kiterjeszjünk ezeket a lehetőségeket?

**V:** Aki szorgalmas olvasója a NetAcademia levelezési listáinak, az már hallhatott a winkey nevű kellemes kis segédeszközről. A program a <http://www.copernic.com/winkey> címről ingyen letölthető, és indítás után csendben megbújik a háttérben.

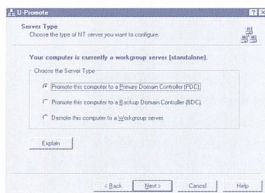


**K:** Ha már a felhasználói felület „tuningolásánál” tartunk: a nemrég letöltött TweakUI indításkor valami lejáró próbaidíszsokról üzenget, és aztán természetesen használatatlan. Hogy járhat le egy ingyenes program?

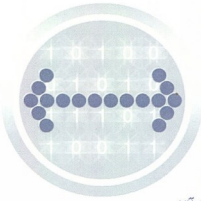
**V:** Ez a hiba a TweakUI egyik bétájában található, bevaltozt dolog. Ahogy kiderült, rögtön meg is jelent egy javított változat, de azzal igazából már nem érdemes sokat foglalkozni, ugyanis azóta megjelent a legújabb, 1.33-as változat is. A Windows ME-n és Windows 2000-n is biztonsággal használható eszköz letölthető a Microsoft weblapjáról: <http://www.microsoft.com/networkstation/downloads/PowerToys/Networking/NTTweakUI.asp>

**K:** Térjünk vissza egy kicsit a Windows NT 4-hez. Igaz az, hogy ha egyszer egy NT 4 Server-t tartománykezelőként (PDC vagy BDC) telepítettünk, abból újratelepítés nélkül nem lehet Standalone kiszolgálót faragni (és viszont)?

**V:** Igen, igaz és nem, nem igaz. Hivatalosan a PDC/BDC/Standalone szerepváltához elkerülhetetlen az újratelepítés, de szerencsére van, aki ebbe a hivatalos álláspontba nem nyugodott bele. A <http://u-tools.com/UTools/UPromote.asp> címről letölthető UPromote nevű ügyes kis eszköz mégis képes erre!



[mICK]



### Web Distributed Authoring and Versioning (WebDAV)

A weben elhelyezett oldalakat megtekinteni könnyű, azonban megváltoztatni azokat korlátozott. Ha az intranetek ilyen lehetőség is rendelkeznek, sokkal hatékonyabbá válna a használatuk.

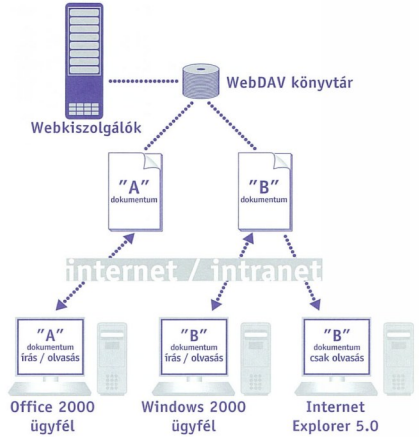
Ennek a hiánynak a pótlására készült a Web Distributed Authoring and Versioning (WebDAV) nevű szolgáltatás, melyet az IIS 5.0 teljes mértékben támogat. A WebDAV lehetővé teszi a szabvány kiterjesztése, mely lehetővé teszi, hogy HTTP-kapcsolaton keresztül bármilyen tároló médium tartalmát kezelhessük (például egy fájlrendszer).

A webkiszolgálónkon felállított WebDAV könyvtáron keresztül a felhasználók megoszthatják dokumentumaikat. A távol dolgozó munkatársak úgy használhatják a nekik kiosztott könyvtárat, mintha az a saját gépükön lenne: fájlokat és könyvtárakat mozgathatnak, kereshetnek, szerkeszthetnek vagy törölhetnek a kiszolgálón.

A WebDAV lehetővé teszi a fájlhozzáférési engedélyek használatát, a kapcsolatmentes (offline) szerkesztést, a fájlok integritásának megőrzését, a párhuzamos változtatásokból eredő konfliktusok feloldását. Az IIS 5.0 WebDAV szolgáltatása a következőket nyújtja:

- ☞ **Az erőforrások kezelése.** A felhasználók módosíthatják a WebDAV nyilvános könyvtárban található fájlokat.
- ☞ **Tulajdonságok megváltoztatása.** A felhasználók lekérdezhetik és módosíthatják az erőforrások tulajdonságait
- ☞ **Erőforrások zárolása és felszabadítása.** Több felhasználó egyidejűleg is olvashat egy fájlt, de egyszerre csak egy módosíthatja azt. Ennek biztonságos megvalósításáról az erőforrások zárolása gondoskodik.
- ☞ **Fájlok tulajdonságai és tartalma szerinti keresés.** A WebDAV könyvtárban lévő fájlokat akár tulajdonságaik, akár tartalmuk szerint kereshetjük. Keresést kezdeményezhetünk például olyan fájlok után, melyek a „tábla” szót tartalmazzák, vagy amelyeknek „Feri” a szerzője. Ugyanezen a szolgáltatáson alapul az Office rovatban ismertetett Office Server Extensions.

A IIS felügyelő eszközei segítségével WebDAV könyvtár felállítása éppoly egyszerű, mint egy virtuális könyvtáré. Miután létrehoztuk a közzététel céljára szolgáló könyvtárat, a megfelelő jogosultságokkal rendelkező felhasználók dokumentumokat tehetnek közzé, és kezelhetik az itt található fájlokat.



### Dokumentumok megosztása a WebDAV segítségével

A WebDAV könyvtárak elérése a Microsoft különféle termékei mellett olyan alkalmazásokkal lehetséges, melyek támogatják az ipari szabványban számító WebDAV protokollt.

- ☞ **Windows 2000-rel az Add Network Place (Hálózati hely hozzáadása)** varázslóval kapcsolódhatunk a WebDAV könyvtárhoz, mely ezután saját gépünk fájlrendszerének szerves részévé válik. A kapcsolódás után fájlokat másolhatunk ebbe a könyvtárba, módosíthatjuk a fájlok jellemzőit, és sok más fájlrendszerre jellemző feladatot is elvégezhetünk.



Windows 2000-ben a Hálózati hely hozzáadása varázsló segítségével létesíthetünk kapcsolatot a WebDAV kiszolgálóval



☞ **A Microsoft Internet Explorer 5** segítségével a fentihez hasonló módon kapcsolódhatunk a WebDAV könyvtárhoz, és ugyanazokat a fájlrendszerbeli műveleteket véghezjuttatjuk el.

☞ **A Microsoft Office 2000** bármely alkalmazása képes a dokumentumokat közvetlenül a WebDAV könyvtárban létrehozni, szerkeszteni, elmenteni és ott közzétenni. Mivel megfelelő engedélyek birtokában bárki írhat a WebDAV könyvtárba, fontos, hogy korlátozni tudjuk annak elérését. Az integrált Windows hitelesítés használatával biztosíthatjuk, hogy csak a megfelelő engedélyekkel rendelkező felhasználók érhessek el az intraneten megosztott WebDAV könyvtárat.

### Webmappák

A WebDAV könyvtárakra, vagy a bennük található állományokra mutató parancsikonokat *webmappáknak*, vagy *HTTP mappáknak* nevezzük. Ezek a parancsikonok a **Hálózatok (My Network Places)** mappában automatikusan létrejönnek, amikor a WebDAV szolgáltatással rendelkező kiszolgálón valamilyen erőforrást nyitunk meg és írási /olvasási joggal is rendelkezünk. Ezen kívül az **Add Network Place (Hálózati hely hozzáadása)** varázslót is felhasználhatjuk a webkiszolgálóra és más számítógépekre mutató parancsikonok létrehozásához.

A webmappák használatával a felhasználók úgy érezhetik, mintha a távoli kiszolgálón lévő könyvtár saját fájlrendszerük részét képeznék. A webmappákat is a Windows-ban megosztott felhasználói felület segítségével érhetjük el, így látszólag nincs különbség a helyi fájlrendszer, hálózati meghajtó vagy egy Interneten lévő webhely tartalmának böngészése között. Egy kis eltérés mégis feltűnhet: a webmappa tartalmának megjelenítései a benne található fájlok internetes címe is megjelenik.

### Az IIS és az elosztott fájlrendszer (Dfs)

Az IIS 5.0 kihasználja a Windows 2000 elosztott fájlrendszerében (*Distributed File System – Dfs*) rejlő lehetőségeket. A Dfs egyetlen névtérben egyesíti a különböző számítógépeken tárolt állományokat, segítségével egyetlen, hierarchikus rendszerben egyesíthetjük a hálózat különböző kiszolgálóin található információkat, megkönyvíthetjük a fájlok elérését. Ha a webhely gyökérkönyvtárának Dfs gyökérkönyvtárat adunk meg, akkor úgy módosíthatjuk a webhely erőforrásainak helyzetét (*például egyik kiszolgálóról egy másikra mozgathatjuk őket*), hogy a webhely HTML fájljaiban egyetlen hivatkozást sem kell megváltoztatni. (*A Dfs-fában a Windows Media szolgáltatások fájljai is tárolhatók.*)

### HTTP tömörítés

A HTTP tömörítéssel a tömörített lapok fogadására alkalmas böngésző meggyorsíthatja a letöltést. Ez azokban a helyzetekben jelenthet komoly előnyt, ahol a sávszélesség korlátozott. A tömörítéssel elérhető sebességnövekedés a weblapok tartalmától, a kiszolgáló teljesítményétől és terhelésétől is függ. Ha a webkiszolgáló főként dinamikus oldalakat generál, vagy a webkiszolgálót működtető gép pro-

cesszorkihasználatára eléri a 80%-ot, akkor nem érdemes a kiszolgálót a tömörítéssel tovább terhelni.

### Újrakezdhető FTP letöltés

A szabványos File Transfer Protocol (*FTP*) szolgáltatás a Windows 2000 Server szerver részét képezi, sőt az IIS 5.0 az újrakezdhető FTP protokollt is támogatja. Ennek köszönhetően a kapcsolat megszakadása esetén a letöltést a szakadás helyétől lehet folytatni.

### Biztonság

A Microsoft együtt dolgozik az Internetes számítógépes közösséggel, hogy elősegítse a szabványok kialakulását és beépítse azokat termékeibe. Az IIS 5.0 a következő biztonsági szabványokat támogatja:

☞ **Fortezza.** Az IIS 5.0-ban újdonság az Amerikai Egyesült Államok kormánya által javasolt szabvány, a Fortezza támogatása. A Fortezza megfelel a Defense Message System biztonsági architektúrájának, biztosítja az üzenetek bizalmasságát, integritását, hitelességét, megváltoztathatatlanságát, és az üzenetekhez, komponensekhez és rendszerekhez való hozzáférés felügyeletét.

☞ **Secure Sockets Layer (SSL 3.0).** Az SSL biztonsági protokollokat széles körben alkalmazzák az Internetböngészők és kiszolgálók a hitelesítés, üzenet-biztonság megőrzése és a bizalmas információk védelme érdekében. Webkiszolgálónk SSL biztonsági modulját beállíthatjuk, hogy ellenőrizze a felhasználók kilétét és kódolja a hálózaton átküldendő adatokat.

☞ **Transport Layer Security (TLS).** A TLS az SSL-en alapul. Képes a felhasználók biztonságos azonosítására és a programozók számára lehetővé teszi, hogy olyan TLS-t felhasználó kódot írhasznanak, mely anélkül képes más folyamatokkal kódolt információk cseréjére, hogy a kódot létrehozó programozónak a többi program kódját ismernie kellene. A TLS-t arra fejlesztették ki, hogy olyan keretrendszert biztosítson, melyet új, nyílt kulcsú és más kódoló algoritmusok is felhasználhatnak.

☞ **PKCS #7.** Ez a protokoll a digitális aláírások, vagy digitális borítékok formátumát határozza meg, melyek az információt biztonságos formában őrzik. Az IIS hitelesítési folyamataiban mindkettőt felhasználja.

☞ **PKCS #10.** Ezt a protokollt a nyílt kulcs tanúsítványok formátumát írja le, melyeket a tanúsítványokat kibocsátó szervezethez (CA) küldenek el.

☞ **Alapszintű hitelesítés.** A *Basic Authentication* a HTTP 1.0 specifikáció része, a jelszavakat Base64 formátumban továbbítja. Ez a módszer rendkívül elterjedt, ipari szabványnak számít a felhasználói nevek és jelszavak továbbításában. A Basic Authentication előnye, hogy a HTTP szabvány része, ezért ez a legismertebb hitelesítő módszer a böngészők között. A probléma csak az, hogy a jelszavakat nem titkosítja, így a hálózati forgalom figyelésével (*melyre könnyen elérhető eszközök állnak rendelkezésre*) könnyen megszerezhető az ilyen módon átvitt jelszavak, ezért ez a módszer csak azokban az esetekben ajánlható, amikor a web



böngésző és a kiszolgáló közti összeköttetés lehallgatásbiztos, például közvetlen kábelkapcsolat, vagy dedikált vonal esetén.

- ☛ **Digest Authentication.** Az IIS 5.0 egyik újdonsága a Digest Authentication, mely ugyanazokkal a lehetőségekkel bír, mint a Basic Authentication, de más módon végzi a bizalmas információ átvitelét. A hitelesítésre szolgáló információ egyirányú kódolási folyamaton (*hashing*) halad át. A folyamat eredményét hash-nek vagy message digest-nek hívjuk, melyből az eredeti szöveget nem lehet visszakódolni. A kiszolgáló kiegészítő információkat is hozzacsatol a jelszóhoz, mielőtt azt kódolná, így senki sem tudja a kódolt jelszót arra felhasználni, hogy segítségével később jogosulatlanul az igazi felhasználónak adja ki magát. Ez a Digest Authentication legnagyobb előnye, hiszen a Basic Authentication esetén a jelszó elfogása után bárki jogosulatlanul használhatta a jelszó eredeti tulajdonosának biztosított szolgáltatásokat. A Digest Authentication proxy kiszolgálókon és tűzfalakon keresztül is működik, és a Web Distributed Authoring and Versioning (*WebDAV*) számára is elérhető. Mivel azonban ezt a lehetőséget a HTTP 1.1-es szabványában definiálták, nem minden böngésző támogatja. Ha olyan böngésző küld kérelmet a Digest Authentication-t igénylő kiszolgálóhoz, mely nem képes e biztonságttechnikai módszer használatára, a kiszolgáló megtagadja az együttműködést, és hibáüzenetet küld vissza. E módszer ma még csak azokban a tartományokban használható, melyeknek tartományvezérlője Windows 2000-alapú, ügyféloldalon pedig Internet Explorer 5.0, vagy újabb böngésző működik.

### Biztonsági elemek

Az IIS 5.0-ban használt öt fő biztonsági elem a következő:

- ☛ **Hitelesítés,** a felhasználók kitélének ellenőrzéséhez.
- ☛ **Tanúsítványok,** a webhelyet azonosító információk biztonságos küldéséhez és fogadásához.
- ☛ **Elérési jogok,** a webhelyen lévő tartalom védelméhez.
- ☛ **Kódolás,** a tartalom megváltoztatásának megakadályozására.
- ☛ **Auditálás,** a biztonsággal összefüggő tevékenységek naplózásához.

### A hitelesség megállapítása

A hitelesítés segítségével megbizonyosodhatunk arról, hogy az, aki el akarja érni a szolgáltatásainkat, jogosult-e belépésre. Az IIS 5.0 a következő hitelesítési módszereket támogatja:

- ☛ **Anonymous Authentication.** Ez a módszer bárki számára belépési lehetőséget biztosít a web- vagy FTP-kiszolgáló nyilvános területeire anélkül, hogy jelszót kérne. Ha az Anonymous Authentication engedélyezve van, az IIS először megpróbálja ezzel a módszerrel beléptetni a felhasználót, akkor is ha más módszerek is engedélyezve vannak. Ha a felhasználó a web- vagy FTP-helyünk nyilvános részére akar belépni, akkor webkiszolgálónk az IUSR\_számitógépnév felhasználói

fíkot rendeli hozzá, ahol a számítógépnév annak a kiszolgálónak a neve, amelyen az IIS fut. Ha több webhelyet működtetünk kiszolgálónkon, vagy webhelyünkön különböző elérési jogokat igénylő területeket tartunk fenn, akkor több anonymous fíkot hozhatunk létre. Minden egyes web- vagy FTP hely, könyvtár vagy akár fájl számára hozunk létre egy külön anonymous fíkot.

- ☛ **Nyílt jelszavas FTP hitelesítés.** Az FTP kapcsolat kialakításához a felhasználóknak be kell jelentkezniük a kiszolgálóra, megadva felhasználói nevüket és jelszavukat. Ha a kiszolgáló nem tudta azonosítani a bejelentkező felhasználót, hibáüzenetet küld vissza. A nyílt jelszavas FTP-hitelesítés nem tartozik a biztonságos módszerek közé, hiszen a felhasználói név és jelszó kódolatlanul utazik a hálózaton.
- ☛ **Anonymous FTP hitelesítés.** FTP-kiszolgálónkat úgy is beállíthatjuk, hogy tegye lehetővé az anonymous elérést. Ekkor az IIS ezt próbálja meg először, akkor is, ha a Basic Authentication is engedélyezve van.
- ☛ **Integrált Windows hitelesítés.** Ezzel módszerrel lehetővé van arra, hogy a felhasználót a Windows-t futtató számítógépek egymás között azonosítsák. Így a felhasználónak csak egyszer kell bejelentkeznie a Windows 2000 tartományba, a további erőforrások eléréséhez nincs szükség további azonosításra.

A korábban NTLM-nek vagy Windows NT Challenge /Response hitelesítésnek nevezett, integrált Windows hitelesítés biztonságát adja, hogy az a felhasználó jelszavát nem küldi át a hálózaton. Ehelyett a felhasználó böngészője kódolt üzenetváltás keretében igazolja, hogy a jelszó érvényes.

A Kerberos v5 a Windows 2000 biztonsági architektúrájának része. Az NTLM-et felváltó Kerberos, mint elsődleges biztonsági protokoll gyors bejelentkezést tesz lehetővé a Windows 2000 tartományba, megkímélve a felhasználót a különféle jelszavak és azonosítók használatától, akár tartományok között is.

Az integrált Windows hitelesítés a Kerberos v5 hitelesítő protokollt és saját Challenge/Response hitelesítő protokollját is használhatja. Ha a kiszolgálón fut az Active Directory és a böngésző képes a Kerberos v5 hitelesítő protokoll használatára, akkor Kerberos, különben a hagyományos NTLM azonosítás használatára kerül sor.

Bár az integrált Windows hitelesítés biztonságos, korláttal rendelkezik: először is csak a Microsoft Internet Explorerrel használható, másrészt nem működik proxy-kapcsolatokon keresztül. Ennek megfelelően legjobb felhasználási területe az intranetes környezet, ahol mind a felhasználó, mind pedig a kiszolgáló ugyanabban a tartományban vannak, és ahol biztosítani tudjuk, hogy minden felhasználó megfelelően fejlett böngészőt használjon.

### Nyílt kulcsú tanúsítványok

Az IIS tanúsítványalapú SSL szolgáltatása egy kiszolgáló-tanúsítványt, egy opcionális ügyféltanúsítványt és számos digitális kulcsot használ. Ezeket a tanúsítványokat a Microsoft Certificate Services-szel vagy egy külső, hitelesítést kibocsátó szervezettől (*Certification Authority - CA*) szerezhetjük be.



### Tanúsítványalapú hitelesítés.

Webszolgáltatónk Secure Sockets Layer (SSL) biztonsági hitelesítést kétfajta hitelesítésre használhatjuk. A kiszolgálóhitelesítés arra való, hogy segítségével biztosítsuk a felhasználókat arról, hogy a webszolgáltató biztonságos (és valóban az, akinek mondja magát), mielőtt ők bizalmas adatokat (például hitelkártyaszámot) adnának meg. Az ügyfélhitelesítés pedig természetesen a felhasználók személyazonosságának megállapítására használható.

### Kapcsolt ügyféltanúsítványok.

Az ügyféltanúsítványokat összekapcsolhatjuk webszolgáltatónk Windows felhasználói fiókjával. Miután létrehozunk és engedélyeztünk egy hitelesítés-fiók kapcsolatot, az ügyfélhitelesítéssel bejelentkező felhasználót a webszolgáltató automatikusan összekapcsolja a megfelelő Windows felhasználói fiókkal. Ezzel a módszerrel anélkül azonosíthatjuk a bejelentkező felhasználókat, hogy basic, digest, vagy integrált Windows hitelesítést kellene használnunk. Akár egyetlen, de több ügyféltanúsítványt is rendelhetünk egy Windows felhasználói fiókhoz. Például ha cégünk számos különböző osztállyal rendelkezik, vagy többféle üzlettel foglalkozik, és mindegyiknek megvan a maga webszolgáltatója, a különböző osztályokhoz tartozó tanúsítványokat hozzákapcsolhatjuk az osztály saját webszolgáltatójához. Ily módon elérhetjük, hogy az osztályok webszolgáltatója csak a hozzá tartozó munkatársak számára lesz elérhető.

### Titkosítás

Miután megszerveztük az információhoz történő biztonságos hozzáférést, az adatokat meg kell védenünk az Interneten történő átvitel alatt is. Az információ (hitelkártyaszám, telefonszám) biztonságos átvitele titkosítással történik. A titkosítással olvashatatlan, értelmezhetetlen formába hozzuk az információt mielőtt elküldjük, a visszakódolással pedig visszanyerjük eredeti formáját. Ehhez a művelethez az IIS az SSL 3.0 protokollt használja fel, mely biztonságossá teszi a kiszolgáló és a böngésző kapcsolatát; az SSL meggyőződik webhelyünk hitelességéről, és azonosítja a felhasználókat. Az efféle kapcsolat kiépítéséhez mind a böngészőnek, mind a webszolgáltatónak rendelkeznie kell megfelelő titkosító lehetőségekkel. Egy biztonságos kapcsolat ideje alatt egy kulcs készül. Mindkét fél ezt az időszakos (session) kulcsot használja az információ titkosítására. A kulcs biztonsági szintjét bitekben méri; minél több bítet használ a kulcs, annál erősebb a titkosítás. Bár a nagyobb bitszámú kulcsot használó titkosítás biztonságosabb, a kiszolgálón több erőforrást fogyaszt. Webszolgáltatónk általában 40-bites kulcsokat használ, de akár 128-bites is lehet, a kívánt biztonság fokától függően. Úgy is beállíthatjuk webszolgáltatónkat, hogy az SSL-t használó kapcsolathoz minimálisan 128-bites időszakos kulcshosszat igényeljen az alapértelmezett 40-bites helyett, bár ehhez a felhasználóknak megfelelő böngészővel kell rendelkezniük. Ha webhelyünkre a Require 128-bit encryption (128-bites titkosítást igényel) beállítást alkalmazzuk, s kiszolgálónk tanúsítványja 128-bitesnél kisebb kulcsokat tartalmaz, akkor webhelyünk elérhetetlenné válik.

### A webhelyek auditálása

A biztonság utolsó bástyája webhelyünk használatának figyelemmel kísérése. Ehhez auditáló házirendeket (audit policies) kell létrehozniuk a könyvtárakhoz vagy fájlokhoz vagy a kiszolgáló eseményeihez. Az auditáló házirendek által készített biztonsági naplók elemzésével juthatunk az illetéketlen behatólók nyomára.

Saját biztonsági naplónk elkészítéséhez beépített Windows segédprogramokat, az IIS 5.0-ba épített naplózó szolgáltatásokat, vagy ASP alkalmazásokat használhatunk. Ha NTFS fájlrendszerrel rendelkezünk, fájl- vagy könyvtárszinten figyelhetjük az elérésre (fájlok olvasása, írása; könyvtár böngészése) tett sikeres vagy sikertelen kísérleteket.

A kiszolgálóval (a webszolgáltatóra történő be- és kijelentkezés, a webszolgáltató biztonsági házirendjének megváltoztatása, a kiszolgáló számítógép leállítása), webhelyel, virtuális könyvtárral vagy fájljal kapcsolat eseményeket is naplózhatjuk, ezt a a Microsoft Management Console **Audit policies** oldalán állíthatjuk be.

### Biztonsági varázslók

A biztonsági beállítások megkönnyítése érdekében az IIS 5.0 három új biztonsági varázslóval rendelkezik:

☞ **A tanúsítvány varázsló** leegyszerűsíti a tanúsítványok kal kapcsolatos felületeit teendőinket (tanúsítványkérelmek készítése, a tanúsítvány életciklusának nyomon követése). A Secure Sockets Layer (SSL) protokoll támogatása iránt egyre növekvő az elvárás, különösen olyan webhelyek esetén, melyek elektronikus kereskedelemmel foglalkoznak, bizalmas üzleti információkat kezelnek. Az új varázsló segít az SSL-lel működő webhely felállításában, ezen kívül e varázsló segítségével egyszerűen alakíthatunk ki SSL-lel működő titkosítást és ügyféltanúsítványokon alapuló hitelesítést. A Web Server Certificate varázsló segítségével egyetlen programban szerezhetjük és állíthatjuk be, újíthatjuk meg kiszolgálónk tanúsítványait.

Csak az Enterprise Certificate Server-ekhez intézhetünk hálózaton keresztül kéréseket kiszolgálótanúsítványok beszerzésére. Az IIS Web Server Certificate varázsló még az ugyanazon a gépen lévő, egyedülálló Certificate Server-t sem fogja felismerni. Mentjük a tanúsítvány-kérelmet fájlba és kapcsolat kivétel (offline) kérešként dolgozzuk fel. Ez a hálózati kapcsolat alapuló, Enterprise Certificate Server-rel intézett kérelmeket nem érinti.

Ha nem hálózaton keresztül igényeljük a tanúsítványokat, akkor el kell mentenünk a kérelmet lemezre, s azt kell a CA-nak elküldeni. Ha megkaptuk a tanúsítványt, elindíthatjuk a varázslót és ott folytathatjuk, ahol a kérés elmentésénél abbahagytuk. Ha a tanúsítványok cseréjét irtuk elő, az IIS addig használja a régi kérelmet, amíg az új tanúsítvány iránti kérelmünkre a CA-tól egy új tanúsítvány nem érkezik.

☞ **Az engedélyvarázsló** végiglépked az engedélyek beállításának lépésén, így könnyítve meg olyan webszolgáltatók telepítését, melyek csak a megfelelő engedéllyel



rendelkezők számára szolgáltatnak információkat. Az engedélyvárásló feladatorientált megközelítést alkalmaz a web és FTP engedélyek, NTFS elérési jogosultságok és hitelesítő sémák beállítására. Kiválaszt-hatjuk azt a beállítást, mely a legjobban megközelíti kívánalmainkat, s a várásló mindent pontosan beállít: a web és FTP és NTFS engedélyeket, illetve a hitelesítő sémákat. A fínomítást természetesen később az IIS beállítási lehetőségeivel mi magunk tehetjük meg. Két beállítási helyzet közül választhatunk:

☞ **Nyilvános webhely.** Ez a leggyakoribb beállítás. A webhely a nyilvános elérés céljából üzemel, a látogatók anonymous hitelesítéssel léphetnek be és minden fájl megtekinthetnek, és elérhetik az ASP alkalmazásokat is. A rendszergazda minden jogot megkap a webhely irányítására.

☞ **Biztonságos webhely.** Ezt a beállítást a vállalati extranetnek, azaz olyan intranetnek használják, melyeket az Internetről is el lehet érni. A webhelyen lévő információ bizalmas természetű, így ez a beállítás Basic, Digest, vagy integrált Windows hitelesítést használ. Csak a jogosultsággal rendelkező felhasználók számára teszi lehetővé az összes fájl és ASP alkalmazás elérését a webkiszolgálón. A fentiekhez hasonlóan a rendszergazda itt is teljes hatalommal rendelkezik.

☛ **A Certificate Trust Lists (CTL) várásló** a CTL-ek beállításában nyújt segítséget. A CTL egy bizonyos könyvtárhoz tartozó megbízható tanúsítványt kibocsátó szervezetek listája. A CTL-lel szabályozhatjuk, hogy mely CA-któl fogadhatunk el tanúsítványokat. Ez különösen az ISP-k számára fontos, akik számos webhellyel rendelkeznek, és mindegyik webhely különböző CA-kkal működik együtt. A CTL-eket csak webhelyekhez használhatjuk, FTP-helyekhez nem.

### Az alkalmazói programok környezete

Az IIS 5.0 nagyobb teljesítményt nyújt a webalapú alkalmazások készítéséhez, mint elődei. Az IIS-en belül működő Active Server Pages (ASP) motor, a Windows 2000 adatelérést lehetővé tevő komponenseivel és komponens-szolgáltatásaival kombinálva jól használható környezetet teremt a webalkalmazások fejlesztéséhez és működtetéséhez.

Az ASP legfrissebb kiadását folyamatvezérlés, hibakezelés, Windows Script Components és más fejlesztések teszik a script-írók és a webes alkalmazások fejlesztői számára még jobban kezelhetővé. Az ASP alkalmazások gyorsaságát a script nélküli ASP, az automatikus teljesítményhangolás, a nagyobb teljesítményű objektumok és a Windows 2000 operációs rendszer fejlesztései növelik.

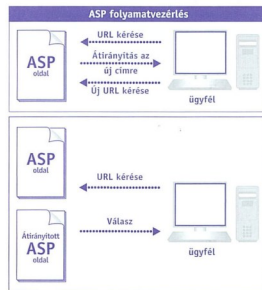
### ASP-újdonságok

Az ASP webes alkalmazások fejlesztésére szolgáló, kiszolgálóoldali scriptkörnyezet, melyel dinamikus, interaktív webes alkalmazásokat hozhatunk létre. Az új folyamatvezérlési és hibakezelési lehetőségek megkönnyítik az alkalmazások fejlesztését és viselkedésük irányítását. A script nélküli ASP feldolgozás és más fejlesztések pedig az ASP oldalak teljesítményét növelik.

### Folyamatvezérlési lehetőségek

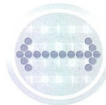
Az új folyamatvezérlési lehetőségek úgy növelik az ASP-alapú alkalmazások teljesítményét, hogy közben csökkentik a kiszolgáló és a böngésző közötti „körutazások” (roundtrip) számát. Átirányítás (Redirect) esetén jelenleg a **Response.Redirect** módszer használatával az oldal elérési idejét egy körút növeli, mivel ez esetben a kiszolgáló nem a kért HTML oldalt, hanem annak csak a címét küldi vissza a böngészőnek. A böngésző ekkor automatikusan elhagyja a kiszolgáló kérelmi sorát, és egy új HTTP-kérelmet intéz a kiszolgálónak az új URL-lel. A kiszolgáló ezt a kérelmet – más böngészők ugyanakkor érkező kérelmeivel együtt – hozzáfűzi kérelmi sorához. Egy erősen terhelt weboldal esetén ezek a körutazások jelentős sávszélességet pazarolnak el és csökkentik a teljesítményt, különösen, ha az átirányítás ugyanarra a kiszolgálóra irányul.

Az ASP Server objektumnak két új metódusa van, melyet a program folyamatvezérlésére használhatunk fel: a **Server.Transfer** és a **Server.Execute**. Ahogy az ábrán látható, a **Server.Transfer** metódust arra használhatjuk, hogy a **Response.Redirect**-tel történő átirányítás helyett a kiszolgálón lévő egyik ASP-fájlról a másikra ugorjunk. A **Server.Transfer** segítségével a kiszolgáló kérelmi sorának elhagyása nélkül térhetünk át az egyik ASP-fájlról a másikra, így megtakaríthatjuk a költséges hálózathasználatot.



**A Server.Transfer-rel megtakaríthatjuk a körutazásokat a kiszolgáló és a böngésző között, ha az átirányított URL ugyanazon a kiszolgálón lévő ASP-oldalra mutat**

Az ASP a **Server.Execute** metódust is a rendelkezésünkre bocsátja, melyet arra használhatunk, hogy a vezérlést átadjuk egy fájlunk, végrehajtsuk a tartalmát, majd visszatér arra a fájlra mely a végrehajtást kezdeményezte. Ha otthon vagyunk a VBScript-ben, akkor a **Server.Execute**-ot eljárás-hívásként képzelhetjük el a legjobban, azzal a kivétellel, hogy ez esetben ASP oldalt hajtunk végre eljárás helyett.



### Böngészőtulajdonságokat tároló komponens

A webalkalmazások során gyakran szükség van a felhasználó böngészőjének és tulajdonságainak pontos ismeretére. A webalkalmazás programozója ezekhez az információkhoz eddig úgy juthatott hozzá, hogy egy ügyfeloldali script segítségével egy cookie-ba írta az adatokat, majd a kiszolgáló kiolvasta onnan azokat.

A Browser Capabilities komponens **BrowserType** objektumát a script felhasználhatja a böngésző tulajdonságainak elemzésére. Amikor a böngésző a webkiszolgálóhoz kapcsolódik, automatikusan elküldi neki a HTTP User Agent fejléct. Ebben a fejlécben a böngésző ASCII karakterláncban definiálja önmagát, nevének és verziószámának megjelölésével. A **BrowserType** objektum összehasonlítja a böngésző által küldött fejléct a `Browscap.ini` fájlban lévő bejegyzésekkel. Ha egyeztet talál benne akkor a fájlban leírt tulajdonságokat rendeli a böngészőhöz, ha nem talál, akkor hozzá hasonlót keres jokerkarakterek („\*“, „?“) használatával. Ha így sem találta meg, akkor a `Browscap.ini`-fájlban beállított, alapértelmezett beállításokat fogja visszaadni. Ha nincs beállítva az alapértelmezett böngésző, akkor az objektum minden tulajdonságát UNKNOWN-ra állít be (*melynek típusa string*).

A `Browscap.ini` fájl szerkesztésével adhatunk új böngészőket a listához vagy módosíthatjuk a régiek tulajdonságait.

### Hibakezelés

Az ASP új hibakezelő lehetőségeivel könnyebben megtalálhatjuk a hibákat webalkalmazásainkban. A hibákat saját ASP-fájllal dolgozhatjuk fel, a **Server.GetLastError** metódussal hasznos információit (*például a hibák leírását, vagy sorszámát*) szerezhethetünk. Az **ASPERror** objektum segítségével egy ASP oldal scriptjében felmerült hiba körülményeiről kaphatunk információkat.

### Script nélküli ASP

A megelőző verziókban minden ASP oldalt úgy kezelt a rendszer, hogy feltételezte: scriptet tartalmaz, s emiatt teljesítményigényes ASP feldolgozást végzett a lapon. Az IIS 5.0 viszont először is megállapítja, hogy az ASP kiterjesztésű fájl tartalmaz-e scriptet, s ha nem, akkor statikus HTML oldalként kezeli, ha igen, akkor átadja az ASP-feldolgozónak. Ezzel az újítással lecsökken a különbség a statikus HTML és a dinamikus ASP fájlok között, hiszen sokkal kevésbé kell ügyelnünk arra, hogy milyen fájlkiterjesztéssel hozzuk létre az oldalakat, az IIS 5.0 gondoskodik arról, hogy az ASP feldolgozás a lehető leghatékonyabb legyen.

### Megnövelt teljesítményű komponensek

Az IIS 5.0-ban a fejlesztők számos népszerű komponent újraírtak, ezzel lehetővé vált az alkalmazások teljesítményének növelése. Ezek az újraírt komponensek gyakori feladatok ellátására szolgálnak, mint például a cserélődő hirdetések (*adrot.dll*), és a böngésző jellemzőinek megállapítására szolgáló komponens (*browscap.dll*).

### Az ASP automatikus finomhangolása

Az ASP jelenlegi verziója képes észlelni, ha egy külső erőforrás blokkolja a kérés végrehajtását. Ha az IIS 5.0 metaadatbázisában az **ASPThreadGateEnabled** tulajdonság engedélyezve van, akkor az ASP új végrehajtási szálakat indít a további kérések kielégítésére és a folyamatok folytatására. Ha a CPU kezd túlterhelődni, akkor az ASP csökkenti a párhuzamos szálak számát. Ez csökkenti az állandó feladatváltásokból eredő terhelést, mely akkor lép fel, ha egyidejűleg túl sok kérést kell egyszerre feldolgozni.

Válaszul a folytonosan változó terhelési állapotokra az IIS a párhuzamosan futó szálak számának szabályozásával éri el, hogy a kiszolgálót érő terhelés mindig optimális legyen. Az **ASPThreadGateEnabled** és más a párhuzamos szálak szabályozásával kapcsolatos tulajdonság alapértelmezése a leggyakrabban előforduló kiszolgálóállapotokra és forgalmi helyzetekre szabott, a beállítások megváltoztatása csökkentheti az IIS hatékonyságát.

Az **ASPThreadGateEnabled** tulajdonság minden, a webkiszolgáló folyamatában és a külön futó, csoportosított folyamatban működő alkalmazásra hatással van.

### XML integráció

Az **Extensible Markup Language (XML)** adatok vagy dokumentumok bonyolult szerkezetének szemantikai leírására alkalmas. Ezeket a dokumentumokat aztán megoszthatjuk különböző alkalmazások, ügyfelek és kiszolgálók között. Az Internet Explorer legalább 4.0-as verzióját használva a hozzáadott Microsoft XML Parser segítségével úgy bővíthetjük ki webkiszolgálónk képességeit, hogy az képes lesz XML-formátumú adatok cseréjére az IE 4.0, vagy későbbi verzióival illetve más XML-feldolgozó képességekkel rendelkező webkiszolgálókkal.

### Windows Script komponensek

Az ASP legújabb változatával arra is lehetőségünk nyílik, hogy az üzleti logikát tartalmazó script eljárásokat újrafelhasználható COM-komponensekké alakíthassuk. Ezeket a webes alkalmazásokban és más COM-alapú programokban felhasználható objektumokat Windows Script Components-nek nevezzük. A Windows Script Components VBScript-tel és az ECMA 262 nyelvi specifikációnak megfelelő más nyelvekkel (*Microsoft JScript® 2.0, JavaScript 1.1*) felhasználásával egyszerű módot kínál COM-komponensek létrehozására és különböző, COM-fogadására képes alkalmazásokban (*IIS, Microsoft Windows Scripting Host*) való felhasználására.

### Kiszolgálóoldali kiegészítések és az SRC attribútum

A kiszolgálóoldali kiegészítések (*server-side includes - SSI*) dinamikus szöveg (*például: aktuális dátum*) beillesztését teszik lehetővé a webdokumentumokba. Az IIS 5.0-val arra is lehetőségünk van, hogy SSI lehetőségeket használjunk a `#include` direktíva helyett ha a HTML **SCRIPT** tagjának **SRC** attribútumát egy kiszolgálóoldali kiegészítés virtuális vagy relatív útvonalnevére állítjuk, és megadjuk a **RUNAT=SERVER** attribútumot.



### Az ASP scriptek kódolása

A HTML-oldalakban lévő scriptek mindig is ki voltak téve a kíváncsi tekinteteknek, azonban az IIS 5 tartalmaz egy új scriptkódoló segédprogramot, melyet a VBScript és JScript 5.0 script-nyelvekhez használhatunk fel. Mind az ügyfél-, mind a kiszolgálóoldali scripteket kódolhatjuk, olvashatatlan ASCII-karakterek halmozásával változtatva az értékes programrészeleteket. A kódolt scriptek futási időben kerülnek visszakódolásra. Bár ez a megoldás nem kimondottan titkosítás, a legtöbb felhasználót megakadályozza abban, hogy elolvassa és lemásolja az oldalakban lévő scripteket.

### Az IIS és a Component Services

Az IIS 4.0-nál a Microsoft Transaction Server tette lehetővé a tranzakciós műveleteket, az IIS 5.0 és a Windows 2000 esetében ezt a munkát a Component Services végzi, számos más, a komponensek fejlesztésével és telepítésével összefüggő feladat ellátásával egyetemben. Az IIS a Component Services alábbi lehetőségeit használja ki:

- ☞ Az alkalmazások különálló folyamatokként elszigetelése.
- ☞ A COM-komponensek közti kommunikáció (beleértve az ASP-ben lévő beépített objektumokat is).
- ☞ A tranzakciók koordinálása a rájuk épülő ASP-alkalmazások számára.

### Összefoglalás

Az IIS 5.0 számos ponton bővíti ki a hagyományos webkiszolgálóról szóló képzeiteinket. Hiba esetén sokkal könnyebb újraindítani a kiszolgálót, így a webhelyek sokkal gyorsabban kerülnek ismét használható állapotba. Az alkalmazások által okozott problémák immár nem jelentenek végzetes veszélyt a webkiszolgálóra. A múltban egy rosszul működő alkalmazás összeomlása akár az egész kiszolgálót magával ránthatta. Ma kiválaszthatjuk, hogy a teljesítményigényes, de biztonságos (minden alkalmazás külön címtérben futtatása) vagy a kevésbé biztonságos de hatékonyabb (a kevésbé fontos alkalmazások csoportjának futtatása egyetlen, közös folyamatban, és csak a fontosak külön folyamatban való futtatása) védelmet választjuk-e. Még nagyobb megbízhatóságot érhetünk el a Windows 2000 Advanced Server által nyújtott fűrtezőszolgáltatás igénybevételével. A méretezhetőség is fejlődött, hiszen most már több webhelyet is futtathatunk ugyanazon a kiszolgálón.

Az IIS telepítését és felügyeletét számos újítás teszi még könnyebbé: Az IIS telepítése immár részévé vált a Windows 2000 telepítési folyamatának. A karbantartás számos módon egyszerűsödött: a felügyeleti eszközöket egyetlen közös helyről lehet elérni, számos lehetőség szolgál a távoli felügyelet elvégzésére, és lehetőségünk van bizonyos feladatok elvégzésével más felhasználókat megbízni. A WebDAV támogatásával az IIS 5.0 lehetővé teszi, hogy a webkiszolgálón dokumentumokat érhesünk el és módosíthatunk is azokat, így a vállalatban belüli együttműködés új szintre léphet.

Az új biztonsági protokollok támogatásával, mint a dígest authentication, Kerberos és a Fortezza, az IIS lehetővé teszi, hogy egyre biztonságosabban használhassuk az Internetet,

és biztonságosan telepíthessünk hálózati alkalmazásokat.

Az alkalmazások fejlesztői minden bizonnyal tudják, hogy az IIS 5.0 a Microsoft Web Server korábbi változatain alapszik, de számos új fejlesztéssel segíti a hatékonyan működő, rugalmas webhelyek és webes alkalmazások kialakítását. A fejlesztések zöme az ASP folyamatvezérlés jobbítására és az ASP oldalak gyorsabb feldolgozására összpontosul.

A Windows 2000 és az IIS 5.0 felhasználásával olyan lehetőséghez jutunk, mellyel mindenféle, az Internetes és intranetes helyekkel kapcsolatos problémát könnyűszerrel megoldhatunk.

### További információk:

A Windows 2000 Server webhelye:

<http://www.microsoft.com/windows/server/>

Bevezető az IIS 5.0-hoz:

<http://www.microsoft.com/windows/server/Overview/features/web.asp>

Az IIS SDK a Microsoft Developer Network Interneten elérhető változatában:

<http://msdn.microsoft.com/library/sdkdoc/iisref/sdkt69f8.htm>

Microsoft TechNet IIS webhely:

<http://www.microsoft.com/technet/iis/default.htm>

A Microsoft Interactive Developer Journal-ban megjelent, „Internet Information Services 5.0” című cikk Interneten olvasható változata:

<http://www.microsoft.com/Mind/0499/IIS5/IIS5.HTM>



Az utóbbi években mindennaposá vált az intranetek használata. Az Office Server Extensions segítségével az intranet egy magasabb szintre léphet – jól működő együttműködési környezetté válhat. Az Office Server Extensions (OSE) a Microsoft Office 2000 felhasználói számára egyszerűbbé teszi a meglévő webtechnológiák használatát, hogy így az intranet olyan együttműködési munkaterületté váljon, amely megkönnyíti a felhasználók közös munkavégzését.

Az Office dokumentumok weben történő közzététele ugyanolyan könnyű, mint a fájlok fájlkiszolgálóra mentése. A megbeszélés (Discussions) segítségével a felhasználók megoszthatják a dokumentum tartalmára vonatkozó elképzeléseiket, a legfontosabb webtechnológiák támogatásával a felhasználók könnyebben megtekinthetik és visszakéreshetik a webkiszolgálókon tárolt dokumentumokat. A marketingigazgató például létrehozhat egy marketingtervet és elmentheti a webkiszolgálóra, így lehetővé teheti, hogy a munkacsoportjába tartozó, más osztályon dolgozó tagok véleményezzék azt, közösen részt vegyenek a továbbfejlesztésében és a vállalati tudás felhasználásával hatékonyan létrehozassanak egy tökéletesített tervet.

#### A munkacsoport web

Az intranethasználat előnye közül a felhasználók általában azt emelik ki, hogy könnyebb a szükséges információk visszakérése és elérése, mint a régebbi módszerek esetében, amelyeknél kinyomtatott adatokat kell kezelni, vagy kiszolgálócímeket kell megjegyezni. Az Internetes technológiákat használó alkalmazások figyelemmel követhetik a webkiszolgálón történő műveleteket, és így például jelezhetik egy felhasználónak, ha egy dokumentum tartalma megváltozott. Ezért aztán természetes, hogy az intranetek a munkacsoportok együttműködésének alapvető tényezőivé válhatnak. Mivel az intranetek sok hivatalos vállalati információt is tartalmaznak (pl. *vállalati irányelveket*), sok szervezet rögzített eljárást dolgozott ki az intranetre történő közzététel szabályozására. Néhány információ esetében ez a szabályozás fontos, de sok információ típus esetében nem szükséges. Az ellenőrzés unalmassá válhat a rendszergazda számára, és szűk keresztmetszette válhat a közzététel során. Ez a régi módszer segítségétlen késélemet is okozhat.

Napjainkban a legtöbb szervezetnek részletesen kidolgozott eljárása van a webtartalom közzétételére. Egy munkatárs létrehoz valamilyen tartalmat (*gyakran Office alkalmazásban*), ez a tartalom a rendszergazdához kerül, aki a dokumentumot más formátumúvá (pl. *HTML-lé vagy PDF-féjé*) konvertálja, elhelyezi egy közzétételi kiszolgálón, majd pedig egy webhelyen. A dokumentum módosítások, még ha csak egy kis sajtóhibáról van is szó, az eredeti szerző nem rendelkezik jogokkal a végső dokumentum szerkesztésére. Végig kell csinálni ugyanezt az eljárást, amely ismét ugyanannyi munkát ad a rendszergazdának. Ez az eljárás szűk keresztmetszeteket hoz létre a közzététel során, amelyek hatással vannak a közzététel időszerezésére. Az eljárás egyáltalán azt is jelenti, hogy a tartalmat nem ismerő személy dönt arról, hogy a tartalom hogy jelenjen meg a weben.

Még ha a szerző rendelkezik is engedélyekkel a közzétett dokumentum szerkesztésére, az Office 2000 és az Office Server Extensions nélküli a következő problémák merülhetnek fel a webdokumentumok közzétételével és a velük való munkával kapcsolatban:

- ☞ Nincs olyan egyszerű módszer, amellyel a felhasználók a webkiszolgálót használhatják.
- ☞ Nincs olyan egyszerű módszer, amellyel a felhasználók dokumentumokat tehetnek közzé egy webkiszolgálón.
- ☞ Nincs olyan egyszerű módszer, amellyel levezethető a dokumentumokkal kapcsolatos megbeszélés.
- ☞ Nincs módszer a webdokumentumokkal kapcsolatos kapcsolat nélküli munkára.
- ☞ Nincs automatikus módszer a tartalom változásának követésére.

Lássunk egy példát:

- ☞ Vállalatunk legnagyobb vevoje ajánlatot kér 150 ezer forint értékű termékre és szolgáltatásra. A projekt vezetésére Józsi jelöljük ki az eladási osztályról, Ádám a pénzügyi osztályról az ajánlat jövedelmezőségéért felelős. Zsuzsa a vevőszolgálatról és Péter a termékfejlesztési osztályról arra ügyel, hogy olyan ajánlat jöjjön létre, amelyet a vállalat teljesíteni is tud.
  - ☞ Józsi először is egy e-mail-t küld az általa, valamint az Ádám, Zsuzsa és Péter által végrehajtandó feladatokról. Péter Józsi irányításával létrehozza az első vázlatot a termékajánlatról, és elmenti azt a Customer Proposal webmappába. Ezután Zsuzsa hozzáírja a szolgáltatásokra vonatkozó részt, majd ismét a webre menti.
  - ☞ Ezután Ádám átnézi ezeket, és pénzügyi analízist készít róluk, majd Józsi az egészet összeszerkeszti. A saját rész írásakor mindenki szerkesztheti és jegyzetekkel láthatja el a többi részt. Ezenkívül főnökeik és munkatársaik is írhatnak hozzá megjegyzéseket és egyéb információkat.
- Az ajánlat készítésének dinamikusnak kell végbemennie, függetlenül attól, hogy Word dokumentum, Microsoft Excel munkafüzet vagy PowerPoint prezentáció formájában készül-e. Az Office Server Extensions-szel az alkalmazottak egyszerűen közzétehetik ezt az ajánlatot vagy bármilyen más dokumentumot a weben, a tartalmat közösen hozzák létre és láthatják el megjegyzésekkel, és értesülhetnek róla, ha az adatok módosulnak. A Microsoft FrontPage 2000-rel a csoport tagjai egyszerűen létrehozhatják és kezelhetik ezeket a webhelyeket.

#### Az Office Server Extensions

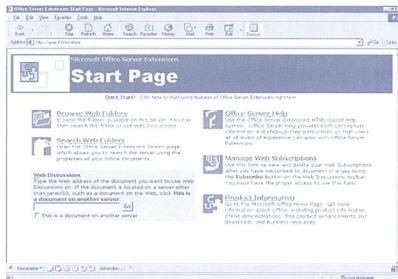
Az Office Server Extensions egy webes alkalmazáskészlet, amely a FrontPage 2000 Server Extensions-re (*amely az Office Server Extensions-szel együtt települ*), az Active Server Pages-re és az ActiveX Data Objects-re (ADO) épül. Az Office Server Extensions az ügyféloldali szoftverekkel működik együtt az Office 2000-ben, a Windows Explorer-ben és a webböngészőben. Az Office Server Extensions többek között az alábbi funkciókat biztosítja a webkiszolgáló számára:



- **Webes közzététel:** A felhasználók a fájlokat webkiszolgálóra menthetik, és könnyen megnyithatják azokat Office 2000 alkalmazásokkal. Olyan egyszerűen használhatják a webhelyet, mint egy fájlmegeosztást.
- **Webes megbeszélések:** A felhasználók úgy vitathatják meg egy dokumentummal vagy annak tartalmával kapcsolatos elképzeléseiket, hogy az Internetböngészővel vagy egy Office 2000 alkalmazással megjegyzéseket írhatnak a HTML, vagy Office formátumú dokumentumokhoz. A felhasználók a megjegyzésekhez is írhatnak megjegyzést és egyszerre is dolgozhatnak.
- **Előfizetés és értesítések (Subscriptions and Notifications):** A felhasználók „előfizethetnek” egy webkiszolgáló megbeszéléseire, dokumentumaira vagy mappáira, és e-mail-ben automatikusan „értesítést” kapnak, ha ezeknek a webdokumentumoknak megváltozik az állapota, így gyorsan reagálni tudnak az új információkra.
- **Keresés:** A felhasználók a webhelyen teljes szövegű keresést is végrehajthatnak, vagy az Office dokumentumok tulajdonságaira (pl. cím, szerző, kategória, kulcsszavak vagy egyedi tulajdonság) is kereshetnek.
- **Kezdőlap:** A kezdőlap egyszerű hozzáférést nyújt az összes Office Server Extensions funkcióhoz, és felületet biztosít azoknak a felhasználóknak, akik nem rendelkeznek telepített Microsoft Internet Explorer 4.0 vagy 5.0 böngészővel, illetve Office 2000-rel.

#### A Kezdőoldal

A Kezdőoldal használatával a felhasználók az Office Server Extensions minden funkciójához könnyen hozzáférhetnek.



#### A Kezdőoldal funkciói:

- **Browse web Folders:** A Kezdőoldal automatikusan listát generál a web alkönyvtárairól. Így a felhasználók megtekinthetik a letölthető és megnyitható fájlok listáját.
- **Search web Folders:** A kezdőoldallal a felhasználók Office tulajdonságok (pl. szerző, cím) vagy kulcsszavak alapján kereshetnek.
- **webServer Help:** HTML alapú Microsoft Sűgő az Office Server Extensions-höz.
- **Product Information:** Hivatkozás a Microsoft Office Update webhelyre, amely további információkat, leltéteket és erőforrásokat tartalmaz.

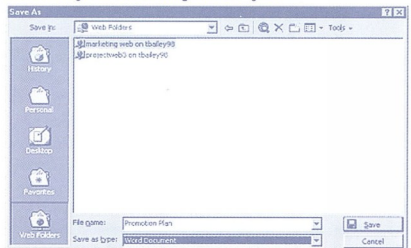
#### Egyszerű webes közzététel

Valószínűleg az Office Server Extensions azon tulajdonságai hatnak leginkább a szervezetre, amelyek megkönnyítik a webes tartalom közzétételét. Ezek a tulajdonságok olyan egyszerűvé teszik a tartalom intranetben történő közzétételét, mint amilyen egyszerű a fájlok elmentése a hálózati kiszolgálóra. Minden alkalmazott képes lesz intranet tartalom létrehozására és közzétételére. Így Péter, az előzőekben említett mérnök, az ajánlat rá eső részét ugyanolyan egyszerűen mentheti el egy webhelyre, mint egy fájlkiszolgálóra, ezzel lehetővé téve, hogy a megfelelő engedéllyel rendelkező felhasználók megtekinthessék azt.

Az itt ismertetett funkciók olyan munkacsoportok számára hozzáférhetők, amelyek telepített FrontPage 2000 Server Extensions-t tartalmaznak vagy WebDAV-ot alkalmazó webkiszolgálót használnak (Microsoft Internet Explorer 5 ügyféllel). Ugyanakkor, ha a munkacsoport webkiszolgálója a FrontPage Server Extensions-t használja, a csoport tagjai további közzétételi funkciók előnyeit is kihasználhatják, így például a dokumentumzáró és helyszínelő eszközöket, vagy a dokumentumokban található hivatkozások ellenőrzését.

#### Webmappák

Az Office Server Extensions a webkiszolgálók kezelésével továbbfejleszti a webes közzétételt. A felhasználók minden olyan hagyományos fájlműveletet elvégezhetnek a webkiszolgálókon, amelyet helyi merevlemez meghajtókon és hálózati kiszolgálókon szoktak. Az Office 2000 alkalmazásokban és a Windows Intézőben a bővített fájlkezelő párbeszédpanelek használatával kevés tanulásal maximálisan kihasználhatják a webkiszolgálók előnyeit.



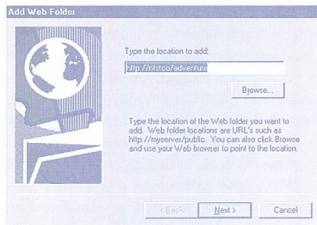
A felhasználók fájlokat nyithatnak meg a webkiszolgálóról, új és módosított fájlokat menthetnek el oda, használhatják a „fogd és vidd” fájlműveleteket, új mappákat hozhatnak létre, törölhetnek és böngészhetnek az elérhető fájlok között.

#### Webmappák hozzáadása

Ezzel a funkcióval egyszerűvé válik új webmappák hozzáadása a felhasználó által közzétételre gyakran használt helyek listájához. A funkció a Fájl megnyitása és a Fájl mentése párbeszédpanelből, valamint a Windows Intézőből érhető el. A Fájl megnyitása és a Fájl mentése párbeszédpanelre kézzel beírt webkiszolgáló címet automatikusan beke-relik a felhasználó webmappa-listájára.



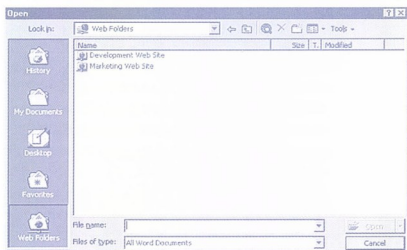
Microsoft  
**Office**



### A Fájll megnyitása

Hogy a webkiszolgálóhoz történő csatlakozás ugyanolyan könnyű legyen, mint a fájlkiszolgálóhoz történő, a fájl megnyitása párbeszédpanel felsorolja azoknak a webkiszolgálóknak a parancsikonjait (*webmappák*), amelyekhez a felhasználó hozzáférést céljából hozzáférhet. A fájlnevé mezőbe bármilyen URL is begépelhető, így a webhely tartalma böngészhető és a kiszolgáló a webmappák felsorolásához adható.

Ezekkel az új közzétételi funkciókkal Józsi és csoportja



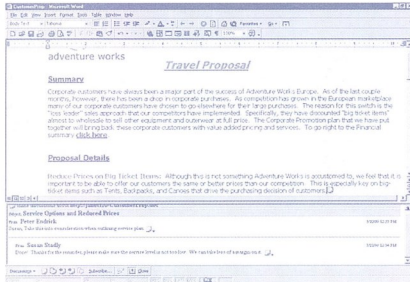
egyszerűen létrehozhatja és frissítheti az ajánlatot, mivel a friss információk bármilyen gyakorisággal könnyedén közzétehető a weben.

### HTML vagy bináris?

Az Office Server Extensions funkciói az eredeti bináris Office fájlformátumokkal, a rich text formattal (RTF) és a HTML-lel működnek. A szerzőnek nem kell azon gondolkodnia, hogy melyik formátumot használja az Office Server Extensions használathoz – a formátum megválasztásának alapja az, hogy milyen alkalmazásfunkciókat igényel a tartalomhoz hozzáférő felhasználó. Az általános szabály az, hogy minél szélesebb körben terjesztjük a dokumentumot, annál fontosabb, hogy az HTML-ben legyen. Másrészt, ha a felhasználók dolgozni is fognak a dokumentummal – nemcsak megtekinthetik azt – és alkalmazás-specifikus funkciókat igényelnek a kezeléséhez, a dokumentumot az Office alkalmazás eredeti bináris fájlformátumában kell hagyni.

### Webes megbeszélések

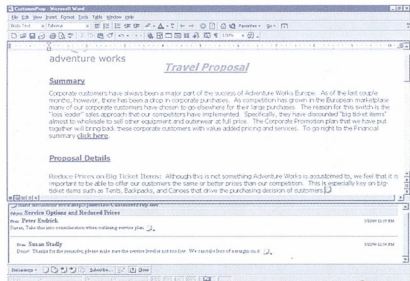
A webes megbeszélések (*Discussions*) az Office Server Extensions új együttműködési funkciója. A megbeszélés során a felhasználók új tartalom létrehozása vagy meglévő tartalom értékelése érdekében megjegyzéseket írhatnak a webdokumentumokhoz.



### A dokumentummal kapcsolatos kérdések megbeszéléssel vitathatók meg.

Miután Péter és Zuzsa már létrehozta a termék- és szolgáltatásajánlatot, Ádám átnézheti azt, és megjegyzéseket, kérdéseket írhat hozzá. A web Discussions használatával ezeket közvetlenül a dokumentumba írhatja be. Péter és Zuzsa ezután átnézheti az Ádám által felvetett kérdéseket.

Mivel a webes megbeszélések megjegyzései egy adatbázisban tárolódnak, az eredeti webdokumentum nem változik meg. A megbeszélés során a munkacsoportok egyszerűen megvitathatják az új programokkal és termékekkel kapcsolatos kérdéseket. A szerző visszajelzést kap a csoporttól az új dokumentum közzététele előtt, majd a közzététel után az eredeti dokumentum megváltoztatásának veszélye nélkül vállalati szinten is kaphat visszajelzéseket. Az is lehetséges, hogy hálózati vagy Internet kiszolgálókon tárolt dokumentumokról folytassunk megbeszélést, de ez adatvédelmi okokból letehető. Így ha például Péter a termékajánlat létrehozása közben meglátogatja egy versenytárs webhelyét és ott talál valami érdekeset, írhat egy olyan megjegyzést, amely a helyi megbeszélési adatbázisban tárolódik. A csoport többi felhasználója akkor is megtekintheti ezt a megjegyzést, ha a versenytárs webhelyére nincs telepítve az Office Server Extensions.



### A dokumentumokkal vagy weboldallal kapcsolatos megbeszélések a webböngészőből is lebonolyíthatók

Ha a webkiszolgálóra telepítve van az Office Server Extensions, a felhasználók a megbeszélést mind eredeti Office dokumentumok, mind HTML fájlok esetében lebonyolíthatják. A megjegyzések az Office 2000 egy speciális eszköztárának használatával mind Office alkalmazásokkal, mind a böngészővel hozzáférhető, vagy akkor is, ha a felhasználó a megbeszélést a Kezdőlapon keresztül éri el. Mint az előző ábrán látható, a megbeszélés a dokumentumon belül vagy átfogó megbeszélésként is megtekinthető, amely a lap alján található Discussion panelen keresztül érhető el. A Discussions eszköztár használatával a felhasználók új megjegyzéseket szűrhetnek be; megtekinthetik, szerkeszthetik és megválaszolhatják a meglévő megjegyzéseket; előfizethetnek egy adott dokumentumra és megtekinthetik vagy elrejtik a Discussions ablakot.

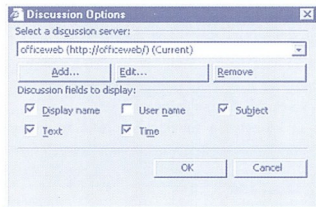
Az Internet Explorer 4.0-ban vagy 5-ben és a Microsoft Word-ben a megjegyzések az alatt a bekezdés alatt jelennek meg, amelyre vonatkoznak. Ha a böngésző Internet Explorer 3.0 vagy Netscape Navigator, a felhasználó a Kezdőlapra lép be, és a kiszolgáló létrehoz egy alsó keretet a megjegyzések számára. Az Office Server Extensions telepítésével a munkacsoportok számára lehetővé válik, hogy tagjait önállóan és akár egyidejűleg dolgozhasanak együtt a webkiszolgáló bármely Office vagy HTML dokumentumán.

#### A web megbeszélések adatbázisa

Az Office Server Extensions a megjegyzéseket a dokumentumoktól külön, egy saját adatbázisban tárolja. Ez az architektúra a következő előnyökkel jár:

- **Megmarad az eredeti dokumentum.** Az eredeti dokumentumhoz ténylegesen semmi sem adódik hozzá és semmi sem változik meg benne. A megbeszélések adatbázisa tárol minden megjegyzést, valamint azt is, hogy az adott megjegyzés a dokumentumon belül hová kapcsolódik.
- **Nem mindenki látja a megbeszélést, aki megtekintti a weboldalt, dokumentumot vagy más tartalmat.** A megbeszélések adatbázisának adatvédelme külön kezelhető a web fájlengedélyeitől. Ez egy speciális, Collaborators nevű felhasználói csoport használatával érhető el, amely a felhasználóknak jogot biztosít a kiszolgáló böngészéséhez és a web megbeszélésekben való részvételhez, de nem ad más jogot a kiszolgáló kezeléséhez. Azok a felhasználók, akik nem rendelkeznek engedéllyel egy adott dokumentum szerkesztésére, így engedélyt kaphatnak a dokumentum megtekintésére és a megbeszélésben való részvételre.
- **A felhasználók egyidejűleg készíthetnek megjegyzéseket egy adott webdokumentumhoz.** Mivel a web Discussion adatbázis többfelhasználós, egy időben több felhasználó készíthet megjegyzéseket egy dokumentumhoz. Mivel maga a dokumentum nem változik, nincs szükség a verziók kezelésére. Egy szervezetben létrehozható egy központi megbeszélési kiszolgáló, így mindenki részt vehet a munkában. Ez lehetővé teszi, hogy a felhasználók kevés adminisztratív munkával vegyenek részt a megbeszélésekben. Ha korlátozni akar-

juk, hogy ki vehessen részt egy adott megbeszélésben, különböző webeken több megbeszélési adatbázisra van szükségünk. A felhasználók a Discussion Options menüjében kiválaszthatják, hogy melyik megbeszélési adatbázist kívánják használni. Egy másik megbeszélésben való részvételhez a felhasználónak egy másik kiszolgálót kell választania. A felhasználói csoportok számára a rendszergazda hozhatja létre ezt a beállítást és az Office 2000 telepítésekor teheti ezt is az Office Profile Wizard használatával. Az Office Profile Wizard-ról részletesen a Microsoft Office 2000 Resource Kit-ben vagy az Office 2000 Deployment and Maintenance leírásban olvashatunk.



Alapértelmezésben a megbeszélések a Microsoft Data Engine-ben (MSDE) tárolódnak, amely egy 100%-osan Microsoft SQL Server kompatibilis adatbáziskezelő. Az MSDE akkor települ, ha semmilyen más SQL Server adatbázis nincs a webkiszolgálóra telepítve.

Ha rendelkezünk telepített SQL Server 6.5-tel (vagy ennél újabb verzióval) az Office Server Extensions-t tartalmazó kiszolgálón, vagy az távolról elérhető a hálózaton, azt is használhatjuk az Office Server Extensions megbeszélési adatok tárolására. A távoli adatbázis eléréséhez az Office Server Extensions-t a „no database” beállítással telepítjük. Így nem települ az MSDE, és a rendszergazdának lehetősége nyílik arra, hogy a telepítés befejezése után más adatbázist válasszon. Ha meglévő SQL Server adatbázist használunk megbeszélési adatbázisként, azt együtt kezelhetjük a többi SQL Server adatbázis adatával. Az MSDE önszabályozó, és rendelkezik egy olyan segédprogrammal is, amellyel az Office Server Extensions elmentheti a megbeszélési adatbázisokat. Ha az MSDE-ről később át szeretnénk térni az SQL Server 7.0-ra, ezt egyszerűen az SQL Server 7.0 telepítésével tehetjük meg. A megbeszélési adatbázis részletezéséért lásd a Microsoft Office 2000 Resource Kit-et.

#### Előfizetés és értesítés

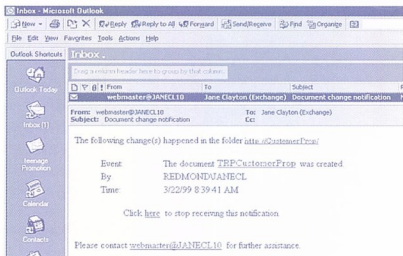
A felhasználóknak tudniuk kell, hogy a legfrissebb tartalommal dolgoznak-e. A programok és a megbeszélések naponta változnak, de mindenkinek ugyanazzal az információval dolgoznia.

Az Office Server Extensions előfizetés (Subscription) funkciójával a felhasználók előfizethetnek egy dokumentumra, mappára vagy megbeszélésre, és így e-mail-ben „értesítést” kapnak, ha az módosul. A felhasználók az értesítés gyakoriságát is meghatározhatják, így idejében értesülnek arról, ha megváltozik egy megbeszélés, fájl vagy mappa tartalma,

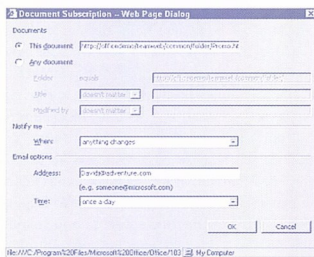


Microsoft  
Office

de nem kapnak túl sok e-mail-t és a dokumentumokat sem kell rendszeresen ellenőrizniük. A csoport tagjai a legfrissebb információhoz jutnak hozzá, a hozzászólások pedig gyorsan és hatékonyan összegyűjthetők. Tegyük fel, hogy Péter, Józsi, Szusza és Ádám mind előfizettek a Customer Proposal webzsupra. Amikor Péter befejezte a termékajánlatot, mindannyian értesítést kapnak arról, hogy elmentette azt a webre, így Szusza azonnal hozzáférhatja a saját részét és a Péterével kapcsolatos megjegyzéseit.



Ez az e-mail üzenet értesíti a címzettet arról, hogy a TRPCustomerProp dokumentum bekerült a CustomerProp webes munkacsoportba. Az üzenet közvetlen hivatkozást tartalmaz a dokumentumra.



### Az előfizetés egyszerű művelet

Asz adatvédelem tervezésénél vegyük figyelembe, hogy a egy webkiszolgálón engedélyezzük az előfizetést, a felhasználók olyan mappákra is előfizethetnek, amelyekhez nem rendelkeznek olvasási engedéllyel, és így értesítést kapnak azok változásairól. Ezzel nem változnak meg a felhasználók fájlengedélyei, de figyelemmel kísérhetik a fájlok módosításait.

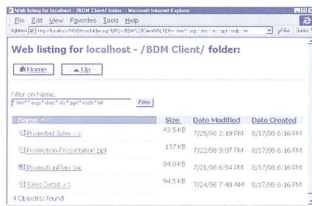
### Dokumentum böngészése, keresése és elérése

Néhány évvel ezelőtt könnyebb volt az Interneten információkat megtalálni, mint a legtöbb szervezetben. Azóta a szervezetek nagy része webes technológiákat használ az információk belső keresésére. Bár így a felhasználók képesek kulcsszavakkal keresni, továbbra sem tudnak egy webmapában böngészni mondjuk a Microsoft Word-ből vagy Excelből, és nem rendelkeznek egységes kezdőhellyel sem. Az

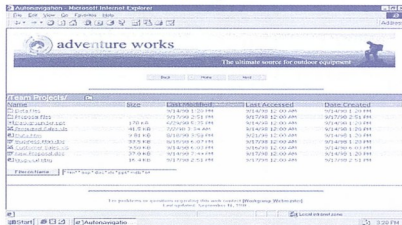
Office Server Extensions a Microsoft Index Server-t használja, és saját maga is telepít egyéb olyan technológiákat, amelyekkel a felhasználók egyszerűen megtekinthetik, kereshetik és kezelhetik a webkiszolgálókat dokumentumait. Ha például Péter, Szusza, Ádám és Józsi munkatársai vagy főnökei meg akarják tekinteni az ajánlatot, és engedéllyel is rendelkeznek ehhez, de nem tálnak rá hivatkozást, a webhelyen megkereshetik azt.

### Az AutoNavigation oldalak

A tipikus HTML dokumentum csatolt objektumokat vagy képeket is tartalmaz. Egy HTML dokumentum mappája egy HTML dokumentumot és akár húsz vagy több képfájlt is tartalmazhat, és ez elriaszthatja a felhasználókat attól, hogy megtekinthessék egy webmappa tartalmát és kiválasszák a szerkesztendő fájlt. Az AutoNavigation oldalak használatával az alábbi nézet segítségével a szerző egyszerűen kiválaszthatja a számára szükséges fájltpusokat, így egyszerűbbé válik a weben már közzétett dokumentumok szerkesztése.



Az AutoNavigation oldalak egyszerűvé teszik a dokumentumok keresését a webhelyen.



Az AutoNavigation oldalak az intranet igényei szerint testreszabhatók.

### Keresés

Az Office Server Extensions egy keresőoldalt is telepít a kiszolgálóra, és azt a Kezdőoldaldhoz csatolja. A keresőoldal a Microsoft Index Server-t használja, így nem használható, ha az nincs telepítve. Ha az Index Server-t később hozzáadjuk, a hivatkozás használhatóvá válik, és a felhasználók hozzáférhetnek a keresési funkcióhoz. A keresőoldal egy ASP oldal, tehát böngészővel használhatjuk. A webkiszolgáló dokumentumait ezután a hagyományos fájlutajdonosságok (pl. név) vagy hagyományos és egyedi Office tulajdonságok alapján is kereshetjük.

Folytatjuk...





A .NET felülettel kapcsolatban nemrégiben azon töprengtem, hogy hogyan lehetne a saját figyel-mét felhívni erre a technológiára, mert sajnos ez is csak annyira érdekli őket, mint a szemé-tszedés – hogyan lehetne a saját valaha is rá-venni, hogy vezérliket írjon a szemétszédés-ről? Szerencsére végül az Oracle megoldotta a problémát, és rövid időre ugyan, de címlapra ke-rült szemétszédési technológiájuk. *(Bill Gates itt az Oracle által felbérlet, de lebukott magányyomozók ténykedésére utal – a ford.)*

A .NET technológia olyan felületet és eszközkészletet jelent, amelynek komponensei részben az Interneten, részben külön-böző hardvereszközökön futnak. Az elsődleges eszköz termé-szetesen továbbra is a számítógép, ezeken futnak a kiszolgá-lószoftverek, ezek a biztonsági rendszer alappillérei. Az in-formációk elérése pedig bármilyen eszközzel megvalósítható. Mostanában mindenki „digitális világban” gondolkodik. Ez egy olyan világ, ahol fényképeinket bárkinek megmutathat-juk, az általunk kedvelt zenéket gyorsan megszerezhetjük – sőt, még fizetni is milyen könnyű a weben! Vagy vegyük pél-dául a videót. Már csak az adatok mennyisége, és a sebés-ségigény miatt is ez az egyik legbonyolultabb adat, amit számítógépeinkkel kezelni kell, mégis nagy szükség lehet ara, hogy a felvételeket könnyen és egyszerűen viihessük el egyik helyről a másikra. Előfordulhat, hogy valaki nem tud megjelenni egy értekezleten – ekkor jól jöhet egy videókame-ra, és később az értekezlet videóanyagát, vagy annak egy részletét el lehet küldeni e-mail-ben. Ugyanezt a felvételt esetleg a későbbiekben vissza lehet keresni, és az általunk érdekesnek tartott részeket vissza lehet nézni.

A jegyzetelést is szeretnénk egyre inkább áttélni digitális formába. A személyi számítógépek egy új formája, a Tablet PC már ennek a jegyében születik – a .NET generáció gyermeke. Ez az az eszköz, amivel végleg vége a papírmunkának, és vég-legesen beléphetünk a digitális világba. De ahhoz, hogy ez megvalósuljon, nagyon fontos, hogy a képernyő tökéletesen olvasható legyen. Ezzel kapcsolatban rengeteg kutatást végez-tünk. Meg kell találnunk azt a felbontást, ami olvashatóság szempontjából az LCD képernyő képét a legközelebb hozza a papír által nyújtott „élményhez”.

A hardveriparban dolgozó partnereinknek köszönhetően az év végére a kezünkben lesz a Tablet PC prototípusa, amely előreláthatólag a következő évben jelenik meg az üzletek-ben. A Tablet PC képernyője egyszerűen lenyitgöző lesz. Az ötlet, hogy számítógépet használjunk papír helyett hason-ló, mint a Power Point használata diafilmm helyett, csak a hatás sokkal drámaibb lesz.

A mi célunk az, hogy az előbb említett dolgok mellett az egész üzleti élet átkerüljön digitális keretek közé. Ha tüze-sebben megvizsgáljuk a napjainkban zajló elektronikus keres-keletet, akkor az derül ki, hogy az úgynevezett e-com-merce üzletkötésekben még mindig jelentős szerep háru-l a hagyományos folyamatokra. Az eladó és a vevő például sok-szor nem az Internet segítségével talál egymásra, vagy pél-dául az üzlet pénzügyi része már nem az Interneten köttetik. Az emberek ugyanígy nem használják az internetes rendelést,

ha a folyamatba bármilyen hiba becsúszhat, például kétséges a termék minősége, késhet a szállítás, vagy túl nagy pénzek forognak kockán – ilyenkor inkább felveszik a telefont. A két világ találkozása és keveredése azonban még nagyobb meg-bizhatatlansághoz vezethet.

Ahhoz, hogy a szolgáltatás teljes menete on-line menjen végbe, sokkal erősebb infrastruktúra szükséges, melynek alapköve a BizTalk Server lehet, mellyel az egész tranzakció menetet lépésről lépésre nyomon tudjuk követni. Így még az is, aki nem ért a számítástechnikához, könnyen átlátja, és képessé válik folyamatok kézbentartására.

Ha ma megnézzük akár egy vállalat, vagy egy kezdő felhasz-náló honlapját, és megvizsgáljuk a kódot, vagy az állomá-nyok kezelésének módját, láthatjuk, hogy bonyolultságuk miatt nem elég hatékonyak a rendszer elvárásaihoz képest. Ez volt az oka annak, hogy elgondolkoztam azon, milyen technikai háttérrel lehetne az alkalmazások megírását le-gyszerűsíteni. A következő kérdésem az volt, hogy milyen módszerrel lehetne a különböző eszközöket összefogni. És ekkor jött képbe a .NET.

Az ötlet nem sokban különbözik a Microsoft 25 évvel ezel-őltti alapötletétől, hisz itt is mindenek a központja a PC. (:-) – a ford.) Az ötlet akkoriban arról szólt, hogy virágzó szoftveripar kiépítéséhez milliányi személyi számítógépre van szükség. A Microsoft alapításának idején rendkívül ke-vesen voltak a szoftveriparban, egy apróska üzletág volt az egész; és ugyan kevés dolgot adtunk el – viszont nagyon drá-gán. Arról álmodoztunk, hogy egyszer majd az informatika óriási iparrággá válik. Nem is számítottunk arra a robbanás-zerű fejlődésre, amit a különböző gyártóktól érkező gépek tökéletesen kompatibilis működése, így az együttműködés kiszélesedése idézett elő. A Basic fejlesztéséről hamarosan áttáltunk operációs rendszerek készítésére, s 1981-ben ki-adtuk az MS-DOS-t. Most, a Windows korában alkalmazások ezrei közül válogathatunk. Ez mind annak köszönhető, hogy ma már az alacsony ár – sok termék filozófia szerint dolgo-zunk. A szoftveriparnak megvan az a jó tulajdonsága, hogy több termék előállítását nem ugyanolyan mértékben növeli a költségeket, mint más iparágakban. Ez alacsonyabb árat generál, ami több termék előállítását jelenti... ez a körfor-gás mozgatja a PC ipart. A számítógépek milliányi össze-kapcsolódásával a forgalom még tovább fog nőni.

A számítógépiparra most óriási figyelem összpontosul. A Micro-soft a múltkor cégvezetőket hívott össze, hogy megbeszélje ve-lük az irodai munka és a webes felületek kezelésének átalaku-lását, és azt láttuk, hogy a vezetőket igencsak érdekli a téma. A PC megjelenésekor csak nagyon kevés ember értette, hogy miről is van szó tulajdonképpen. Manapság ha kinyi-tunk egy újságot, vagy bekapcsoljuk a tévét, másról se ha-lani. Mára a számítógép abszolút a média részévé vált, s in-nen csak egy lépés, hogy az életünk részévé váljon.

# A legjobbakat tanítjuk



Microsoft  
**BackOffice** 

Microsoft  
**Exchange Server**  
Microsoft Back Office Family Member

Microsoft  
**.net**



**msdn**



Microsoft  
**BackOffice** 

Microsoft  
**SQL Server**  
Microsoft Back Office Family Member

Hivatalos Microsoft oktatásokkal (*Windows 2000, SQL Server IIS...stb.*) és egyedi tanfolyamokkal (*fürtözött rendszerek, hálózatzbiztonság, Active Directory „mélyvíz”...stb.*) a NetAcademia Kft. lehetőséget nyújt Önnek és munkatársainak, hogy lépést tartson a legfrissebb szoftverújdonosságokkal.

Tanfolyamainkon magasan képzett szakemberek segítségével sajátíthatják el a legújabb szoftverek „csinját binját” a legjobb informatikusok, tehát Önök! Tematikus levelezési listánkon, a <http://www.netacademia.net> címen segítségül hívhat több száz szakembert, hogy kérdéseire közösen megtalálják a legjobb megoldást.

Maradjon a legjobbak között!

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

Jelentkezzen tanfolyamainkra a

[tanfolyam@netacademia.net](mailto:tanfolyam@netacademia.net) címen.



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

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

0 0 1 1 1 0 0 1 0 0  
1 1 0 0 1 1 0 0 1 1  
0 1 1 0 0 1 1 0 0 0  
0 0 1 0 0 1 0 1 0 1  
0 0 1 1 1 0 0 1 0 0  
1 1 0 0 1 1 0 0 1 1  
0 1 1 0 0 1 1 0 0 0  
0 0 1 0 0 1 0 1 0 1  
0 0 1 0 0 1 0 1 0 1  
0 0 1 1 1 0 0 1 0 0

2000.10.