Microsoft® TechNet

100% technológia ▪ 0% marketing

Az ADAM címtár használata

Az ASP.NET Cache és az SQL Server 2005 együttműködése

VPN karantén

Miért kell IIS a tűzfalra?

Bevezetés a WMI (Windows Management Instrumentation) használatába

ISSN 15865185 03

9 771586 518005

VI./3. szám
2005. június

MÉG NE MENJEN EL SZABADSÁGRA!

BIZTOSAN INDULÓ NYÁRI KÉPZÉSEK

Microsoft GOLD CERTIFIED Partner | Learning Solutions

Csatlakozzon be! Amíg a szabad helyek tartanak!

Windows alkalmazásfejlesztés .NET-tel (2555/2565)	
Június 6-10.	175.000 Ft/fő
SQL 2000 adminisztráció, üzemeltetés (2072)	
Június 13-17.	169.000 Ft/fő
Windows 2003 hálózat rendszertervezés (2278)	
Június 13-17.	169.000 Ft/fő
Microsoft CRM testreszabás (8032)	
Június 13-14.	119.000 Ft/fő
Microsoft CRM testreszabás, fejlesztés (8308)	
Június 15-17.	169.000 Ft/fő
Webes alkalmazásfejlesztés ASP .NET-tel (2310)	
Július 11-15.	175.000 Ft/fő
Windows 2003 Active Directory üzemeltetés (2279)	
Július 11-15.	169.000 Ft/fő
Windows XP rendszergazdáknak + SP2 (2285)	
Július 14-15.	95.000 Ft/fő
Windows 2003 AD / hálózat infrastr. tervezés (2282)	
Augusztus 1-5.	169.000 Ft/fő
Exchange Server 2003 üzemeltetés (2400)	
Augusztus 8-12.	169.000 Ft/fő

Hozza el kollegáját is!

Rendeljen meg több tanfolyamot egyszerre, vagy hozza el kollegáit a fenti képzésekre! Nyári akciónk keretében további 10-25%-os kedvezmény a feltüntetett listaárakból! SA oktatási utalvány beváltás és szakképzési hozzájárulás igénybevétel!

%

Tovább információk:
SZÁMALK Oktatási Rt. Továbbképzés, 1115 Budapest, Etele út 68.
Simon Ferenc, 203-0304/4122 mellék, simonf@szamalk.hu
www.szamalk.hu/tisza

SZÁMALK

Partner az oktatási megoldásokban

Microsoft TechNet

TechNet Magazin
VI. évfolyam, 3. szám
2005. június

Szerkesztőség és kiadó:
Microsoft Magyarország Kft.
1031 Budapest, Graphisoft park 3.

Felelős kiadó:
Székely Tamás marketingigazgató

Szerkesztő:
Takács Gitta (Epszilon Press)

Szaklektor:
Fóti Marcell (Netacademia)

Lapterv és nyomdai előkészítés:
Dobák Ildikó
(Ars Luna Bt.)

Borítófotó:
Archív

Nyomda:
AduPrint Kiadó és Nyomda Kft.
1033 Budapest, Csikós utca 8.
Felelős vezető: Tóth Béláné

Webcím:
www.microsoft.com/hun/technet/
E-mail:
technetmagazin@microsoft.hu

ISSN 1586-5185

Az MVP-közösség

Nyílt Napok voltak Prágában, a Microsoft 100 MVP (Most Valuable Professional) szakértőt látott vendégül a kelet-európai, közel-keleti és afrikai régió 28 országából.

Mint ismeretes, az MVP címet a Microsoft azoknak a szakértőknek ítéli oda, akik elméleti tudásukat és gyakorlati tapasztalataikat önkéntesen és aktívan megosztják a felhasználókkal, a világ különféle online és „offline" közösségeivel.

A Microsoft az 1990-es évek elején indította el az MVP Awards Programot, hogy elismerje a nagyközönség azon tagjainak munkáját, akik idejüket és tekintélyes műszaki szakértelmüket a többi felhasználó megsegítésére, támogatására fordítják. A címmel rendelkező szakértők között vannak publikáló szerzők – cikkeiket rendszeresen olvashatják például itt, a TechNet Magazin hasábjain is –, Microsoft-termékhez kapcsolódó webhely üzemeltetők, nyilvános előadók, oktatók és professzionális fejlesztők is. Világviszonylatban több mint 2600 MVP létezik, akik 81 országot és 32 nyelvet képviselnek. Kelet-Európában 65-en nyerték el a megtisztelő címet. Az MVP program Magyarországon 2003 nyara óta működik, jelenleg hét MVP címmel rendelkező szakértőnk van.

A prágai program során a résztvevők különböző technikai és fejlesztői témájú előadásokat hallgathattak meg, többek között olyan témákról, mint a mobil technológiák terjedése, a Visual Studio 2005, az Exchange Server, a Cluster technológiák, illetve az Internet Explorer 7.0 újdonságai. A program keretében sor került a Culminis nevű nemzetközi kezdeményezés bemutatására is, amelynek célja az MVP-k és a megoldásszállítók együttműködésének kialakítása.

S hogy mit jelentett a magyar MVP-knek a prágai találkozó? Íme két vélemény:

„A szakmai tartalom, és a számos technikai újdonság mellett az esemény legnagyobb jelentőségét abban látom, hogy egy kicsit kimozdította a résztvevőket abból az internet központú „online világból", amelyben nap mint nap élünk, és lehetőségünk nyílt valós emberi kapcsolatok kialakítására. Ezentúl az e-mail címek, becenevek sokkal többet mondanak majd számomra, hiszen azokhoz egy arc, egy ember is társul majd." (Subicz Péter, Windows Server System – Exchange Server MVP)

„A Nyílt Napokon lehetőségünk volt személyesen találkozni és beszélgetni redmondi Visual Studio fejlesztőkkel, akik bemutatták, jelenleg hol tart a termék a fejlesztésben, mi várható a következő Beta változatban, és előreláthatólag mi az, amit a végleges Visual Studio 2005 fog tudni. Az esemény óriási lehetőség volt számunkra, hiszen közvetlenül a Microsoft szakembereitől kaphattunk részletes információkat." (Soczó Zsolt ASP.NET MVP)

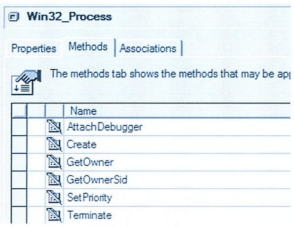
http://www.microsoft.com/mvp
http://www.microsoft.com/communities/
http://www.culminis.com/

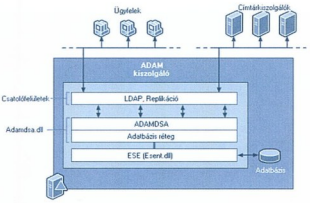
Tartalom

WMI Scripting

Bevezetés a WMI használatába

Remote shutdown, DNS rekord hozzáadása, service újraindítás, netán share létrehozás, mindez távolról, kódból és nagyon egyszerűen? A válasz: WMI!





Az ADAM címtár használata I.

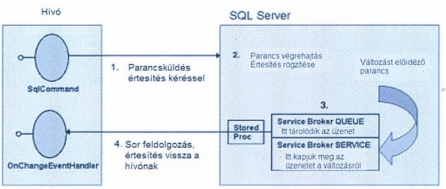
ADAM belülről

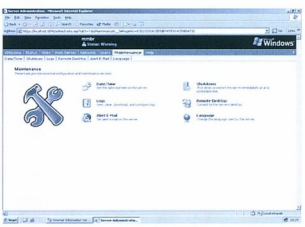
Nagyjából másfél évvel ezelőtt jelent meg a TechNet Magazin oldalain az ADAM címtárról szóló első cikkem, amelyben általánosságban mutattam be az akkor még újdonságnak számító szoftvert. Azóta ADAM kicsit idősebb lett (bár új verzió nem jelent meg), született néhány olyan alkalmazás, amely ezt a címtármegoldást is használja, itt az ideje, hogy részletesebben is megismerkedjünk vele.

ASP.NET 2.0 (Whidbey)

Mi várható a 2005. évi ASP.NET-ben? IV. rész

Az ASP.NET Cache és az SQL Server 2005 együttműködése





Windows szolgáltatások 7. rész

LocalSystem a mi várunk

Újra tekintélyes mennyiségű Windows Server 2003 szolgáltatás működését, jellemzőit ismertetjük meg sorozatunkban.

IT folyamatok a hazai gyakorlatban

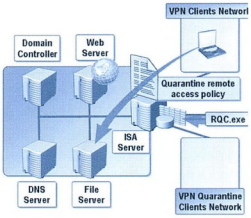
ÉRTÉKELÉS

Az elmúlt években sok (és sokféle) vállalatnál megfordultam, és azt tapasztaltam, hogy nemcsak az egyes szervezetek adottságai térnek el egymástól, hanem azt is, ahogyan ezeket kihasználják, ahogyan élni tudnak a lehetőségekkel. Idővel rá kellett jönnöm, hogy tökéletes vállalat nincs, mindig, mindenhol van mit javítani.

VPN karantén

WINDOWS SERVER 2003 + ISA SERVER 2004 KÖRNYEZETBEN

A VPN csatlakozások számának növekedésével a VPN karantén alkalmazása egyre többször kerül szóba.



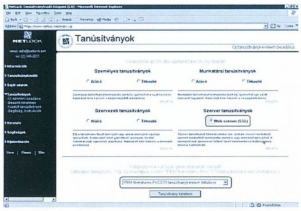
Ami a hivatalos Microsoft tanfolyamokból kimaradt...

ACTIVE DIRECTORY – WEBES REGISZTRÁCIÓ (1. RÉSZ)

Rovatunk e cikkében olyan weblapot készítünk, melynek segítségével a felhasználók maguk kérvényezhetik hozzáférési szándékukat a rendszerünkhöz, intranetünkhöz. Tehát az önregisztráció alapján automatikusan létrehozzuk a felhasználó account-ját, és szükséges beállításait az AD-ben. Továbbá felépítjük az ellenőrzési, jóváhagyási struktúrát úgy, hogy mindez egy webes felületen könnyedén legyen követhető minden szereplő számára.

Dr. Watson

SSL-TANÚSÍTVÁNYKÉRÉS ISA SERVER SZÁMÁRA AVAGY MINEK NEKEM IIS A TŰZFALRA?



Ebben a cikkben leírom, hogyan lehet megvalósítani egy külső hitelesítés-szolgáltató által kibocsátott tanúsítvány segítségével a biztonságos (SSL) kapcsolatot az ISA Server 2004 és a külvilág között.

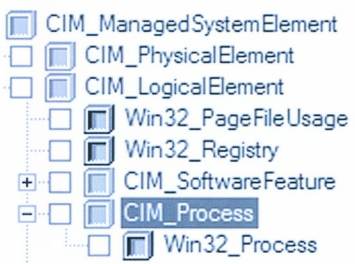
WMI Scripting

Bevezetés a WMI használatába

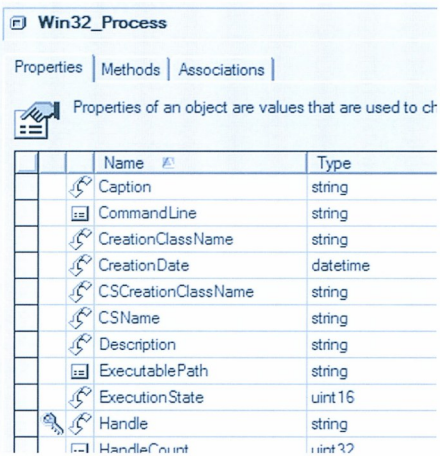
Remote shutdown, DNS rekord hozzáadása, service újraindítás, netán share létrehozás, mindez távolról, kódból és nagyon egyszerűen? A válasz: WMI!

A WMI (Windows Management Instrumentation) célja egy olyan szabványos interface megvalósítása, amely könnyűvé és egységessé teszi a számítógépes rendszerek menedzselését. A WMI a WBEM (Web Based Enterprise Management) szabvány Microsoftos megvalósítása, amelyet a DMTF (Distributed Management Task Force) nevű független társaság definiált. A társaság tagjai többek között a piac vezető nagyvállalatai. A WBEM szabvány egy általános nagyvállalat szoftver és hardver eszközeinek egységes menedzsment-követelményeit definiálja.

A standard több más definíciót is magába foglal, amelyek közül a legfontosabb a CIM (Common Information Model), amely osztályok hierarchiájával és különféle kapcsolataival írja le egy általános számítógéprendszer szerkezetét. Minden osztály egy eszközt/területet reprezentál. A CIM több mint 300 osztálya tartalmazza pl. a CIM_LogicalDisk osztályt, amely egy filesystem partíciót reprezentál, a CIM_Fan osztályt, amely a processzorhűtő ventilátor, a CIM_Process osztályt, amely egy futó processz, és még sok-sok más osztályt. A DMTF csapat úgy készítette el a CIM szabványt, hogy az valóban lefedje egy számítógépes rendszer minden elképzelhető komponensét, viszont ne legyen operációs rendszer specifikus, tehát ne legyen pl. CIM_Registry osztály, amely a Windows operációs rendszerre jellemző. A CIM szabvány az objektum orientáltság lehetőségeit kihasználva, a különböző osztályok között öröklődési kapcsolatokat definiált. Ennek köszönhetően egy jó adag absztrakt osztályt is létrehoztak, mint például a CIM_LogicalElement. A következő ábra a CIM_Process osztályt, annak őseit és gyermekeit mutatja be:



Ne feledjük el, hogy a WMI a WBEM szabvány Microsoftos implementációja, így a CIM szabvány Microsoftos megvalósításának akarva-akaratlanul is tartalmaznia kell különböző Windows-specifikus osztályokat, pl. a registry, a hálózati megosztások és az eseménynapló bejegyzéseinek kezelésére. Az előbbi ábra bemutatta, hogyan valósítja meg a WMI a CIM_Process osztályt: az öröklődési fában végül Win32_Process lesz belőle: az ebből az osztályból létrejött példányokat fogjuk lekérdezni, ha a rendszeren futó processzekkel szeretnénk dolgozni. Hogy miért nem jó nekünk a CIM_Process? A válasz egyszerű: A CIM_Process osztály a CIM szabvány által definiált tulajdonságokat és metódusokat tartalmazza, amely nem teljesen fed le egy Windows-on futó processzt: kell még neki pár tulajdonság, ami nincs benne a szabványban, pl. HandleCount, ExecutablePath és PageFileUsage, amelyek teljesen Windows-specifikus tulajdonságok, tehát leszármaztatjuk a CIM_Process osztályt, és kiegészítjük még pár Windows-specifikus tulajdonsággal, és már meg is van a Windows-os processz osztályunk. Ugyanez a helyzet a további jó pár osztály nagy részével.

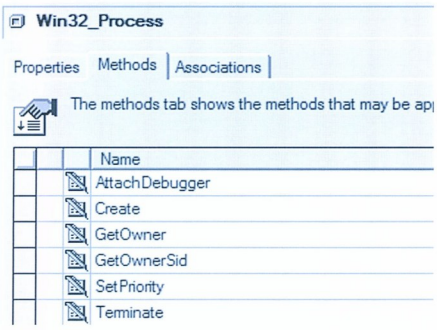


A WMI használata lényegében ezekkel az osztályokkal és a belőlük készített példányokkal történő játék. Az egész dolog nagy előnye, hogy nem kell ismernünk pl. a processzek ke-

zeléséhez, vagy share-ek készítéséhez/lekérdezéséhez /megszüntetéséhez szükséges API-kat, elég csupán egy COM library használatát elsajátítanunk, amelyen keresztül hozzáférhetünk a WMI adatbázishoz. A WMI elérését biztosító COM library-n kívül a .NET framework is lehetőséget nyújt a szolgáltatás elérésére, amelyet szintén bemutatok.
A WMI minden 32 bites Windows-on elérhető. A WMI pontosan úgy használható távolról, mint helyileg. A WMI a legjobb barátod. ☺

Osztályok/példányok/ metódusok/tulajdonságok

Ahogy az az előbbi mondatokból kiderült, a WMI programozása nem más, mint osztályok és azok példányainak lekérdezése, különböző tulajdonságok kiolvasása és metódusok meghívása. Vegyük példaként a már jól ismert Win32_Process osztályt, amely az operációs rendszeren futó user módú processzt reprezentálja. Az osztály pár tulajdonságát már láthattuk az előző ábrán.
Ha szeretném megtudni, hogy milyen processzek futnak az oprendszeremen, futtatok egy WMI lekérdezést, amely lekérdezi az összes Win32_Process típusú objektumot. A lekérdezés eredménye az összes futó processz (Win32_Process példányok) összes WMI-on keresztül elérhető tulajdonsága lesz (lásd az előbbi táblázatban). Ahogy a táblázatban láthattuk, a processz osztály egyedi azonosítója a Handle mező, amely lényegében a processz azonosítója. A Handle elnevezés kompatibilitási okokból maradt (a CIM_Process CIM definíció a Handle mezőt definiálja egyedi azonosítóként). Ugyanez az ID megtalálható a ProcessId mezőben is (ami a Win32_Process osztály saját tulajdonsága).
Ha szeretném leállítani az 1560-as azonosítójú processzemet, nincs más teendőm, mint lekérdezni az 1560-as azonosítójú Win32_Process WMI objektumot, és meghívni annak Terminate metódusát. A következő képernyőkép a processz osztály metódusait tartalmazza:



Ugyanilyen egyszerű egy Windows service (Win32_Service class) elindítása: lekérdezem a service-emet a neve alapján, és meghívom annak StartService metódusát.

Kapcsolódás

Ahhoz hogy hozzáférhessünk a WMI szolgáltatásaihoz, először kapcsolódnunk kell a WMI service-hez. A kapcsolódás nagyon egyszerű, hasonlít egy ADO-SQL adatbázis kapcsolat létrehozásához:

```
Set objLocator =
↳ CreateObject("WbemScripting.SWbemLocator")

Set objSevices =
↳ objLocator.ConnectServer("dszabotitanium")
```

Az előbbi példa a dszabotitanium nevű géphez kapcsolódik, használva a domain-be bejelentkezett felhasználóm jogosultságait. Ezen felül lehetőség van explicit felhasználónév/jelszó megadására is, amelyek a hálózati konfiguráció és a kapcsolódási paramétereknek megfelelően vagy Kerberos vagy NTLM használatával jutnak el a szerverhez. A WMI biztonságtechnikailag teljesen integrált a DCOM-mal, ugyanis minden WMI kérés DCOM-on keresztül továbbítódik, ami viszont RPC-n, mindez azt jelenti, hogy a legújabb operációs rendszerek/biztonsági frissítések (Windows 2003 vagy Windows XP SP2) alkalmazása esetén a WMI szerveren külön oda kell figyelnünk a különböző TCP portok nyitására és DCOM kommunikáció engedélyezésére. Mindezekről később részletesebben írok.
Ott tartunk tehát, hogy bekapcsolódtunk a WMI szerverünkre. Ahogy a példából láthattuk, ha COM-ot szeretnénk használni a WMI elérésére (VB6-ból, VbScript-ből, C++-ból vagy bármilyen COM-kompatibilis nyelvből), a WbemScripting Library-t kell használnunk (%SystemRoot%\system32\wbem \wbemdisp.tlb). Ha .NET-et szeretnénk használni, a System.-Management namespace-t használjuk.

A WMI Scripting COM library főbb osztályai:

- SwbemLocator: a kapcsolódáshoz használatos osztály
- SwbemServices: a kapcsolatot reprezentáló osztály. Ezen az osztályon keresztül tudunk lekérdezéseket futtatni, stb. (ADODB.Connection megfelelője)
- SwbemObject: egy osztályt vagy osztály példányt reprezentál (ADODB.Record megfelelője)
- SwbemObjectSet: lekérdezésnél az eredményt ilyen objektumként kapjuk (ADODB.RecordSet itteni megfelelője)

Sikeres kapcsolódás esetén egy SwbemServices objektumot kapunk vissza, amin keresztül pl. lekérdezhetjük a rendszerben futó összes processzünk összes mezőjét:

```
Dim objProcesses as WbemScripting.SwbemObjectSet
Set objProcesses = objServices.ExecQuery(
↳ "SELECT * FROM Win32_Process")
```

A lekérdezés nyelve ördögien hasonlít az SQL-ben megszokotthoz. Úgy mint az SQL, ez is az ANSI SQL szabványnak megfelelő, és a WQL (WMI Query Language) névre hallgat. A lekérdezésünkben a WMI osztályokat fogjuk fel adatbázistáblákként, ahol a különböző példányok egy-egy rekordként vannak jelen. Minden példányt egyedi azonosítója teszi egyedivé. Fontos, hogy a lekérdezésünket ne túl pazar módon állítsuk össze, ugyanis akárhogy is futtathatunk SQL-hez hasonlító lekérdezéseket a WMI adatbázisban, a WMI mégsem SQL Server. Az adatok lekérdezése „on the fly" történik, azaz a WMI service (amely Windows service-ként fut a szerveren) a kérést az adott WMI osztály provider-éhez továbbítja, amely válaszként előállítja az eredményt, amely bizonyos esetekben költséges is lehet. Tehát mindig figyeljünk oda, milyen mezőket kérdezünk le, ugyanis lehetnek olyan mezők, ame-

lyeknek előállításához hosszú másodpercekre is szükség lehet. Miután megkaptuk az eredményt, fűzzük össze egy string-be, és jelentsük a képernyőn:

```
Dim objProcess as WbemScripting.SwbemObject
Dim strSumProcesses as String

strSumProcesses = "Processzek száma: " &
↳ objProcesses.Count & vbCrlf & vbCrlf

For Each objProcess In objProcesses
   strSumProcesses = strSumProcesses &
   ↳ objProcess.Caption & vbCrlf
Next

MsgBox strSumProcesses
```

Mindez C#-ban a következő képen nézne ki:

```
// Connection point és query összeállítása
ManagementScope scope = new
↳ ManagementScope(@"\\dszabotitanium ");

ManagementObjectSearcher searcher = new
↳ ManagementObjectSearcher(scope, new
↳ ObjectQuery("select * from Win32_Process"));

// Lekérdezés futtatása
ManagementObjectCollection result =
↳ searcher.Get();

Console.WriteLine("Processzek száma: " +
↳  result.Count + "\n");

// Eredmény végiggörgetése
foreach (ManagementObject process in result) {
   Console.WriteLine(
   ↳ process["Caption"].ToString()); }
```

Vissza a COM library-hoz: kapcsolódáskor egy SwbemServices típusú objektumot kaptunk, amelynek az ExecQuery metódusával futtattuk a lekérdezésünket. Ha viszont csak egy objektumot szeretnénk visszakapni, amelynek tudjuk az egyedi azonosítóját, felesleges lekérdezést futtatnunk. Erre való a Get metódus, amelyet a következőképpen tudunk hasznosítani:

```
Dim objProcess As WbemScripting.SwbemObject
Set objProcess =
↳ objServices.Get("Win32_Process.Handle=1512")
```

A Get metódus futásának eredménye az lesz, hogy visszakapjuk az 1512-es azonosítójú processzt egy SwbemObject formájában. Fontos megjegyezni egy újabb hangsúlyos SQL-től való szerkezeti eltérést: míg az SQL adatszerkezetek az adatbázis motor optimalizált működéséhez készülnek, addig a WMI valós eszközök modellezésére született, így előfordulhat, hogy bizonyos osztályoknak két vagy több egyedi azonosító mezője is van. Ilyenkor ezeket együtt kell használnunk. Példa erre a Win32_Product osztály, amely egy telepített szoftver terméket reprezentál. Ennek az osztálynak 3 egyedi azonosítója van: IdentifyingNumber, Name és Version. Egy terméket csak ezzel a 3 azonosítóval együtt tudunk egészen biztosan egyedileg azonosítani. A kulcsmezők definíciója az osztály többi mezőjével együtt az osztály definíciójában szerepel.

A következő példa az előbb lekérdezett processzünket állítja le:

```
objProcess.Terminate
```

Ilyen egyszerű. Ha a metódusnak paramétere is van, a szokásos módon, a következőképpen hívjuk:

```
objProcess.SetPriority(22)
```

Ha egy metódustól visszatérési értéket is várunk, a szokásos módon kezeljük:

```
lngTerminateResult = objProcess.Terminate
```

Fontos megemlíteni, hogy a tulajdonságok/metódusok lehetnek statikusak is, ami azt jelenti, hogy a tulajdonság/metódus nem az osztályból létrehozott példányokon szerepel, hanem magán az alaposztályon. Ilyen például a Win32_Process osztály Create metódusa, amely új processz indítására szolgál. Ezt a metódust nincs értelme egy futó processz példányon meghívnunk:

```
lngReturnValue =
↳ objServices.Get("Win32_Process").Create(
↳ "C:\WINDOWS\System32\notepad.exe")
```

Monikerek

WMI esetén is lehetőségünk van monikerek használatára. Akinek nem tiszta, a moniker a következőkből áll: egyfelől egy shortcutból (a registryben), amely COM objektumok egyszerűsített létrehozására szolgál, másfelől egy sereg paraméterből, amit a COM objektumoknak fel kell dolgozniuk. A következő sor kód egy WMI kapcsolatot hoz létre, és lekérdezi a Terminal Service-t reprezentáló Win32_Service instance-t:

```
Set objTermService =
↳ GetObject("WinMgmts://dszabotitanium
↳ /root/cimv2/Win32_Service.Name='TermService'")
```

Példánkban a WMI moniker szintaxisa a következő:

```
WinMgmts://Szervernév/Namespace/Wmi
↳ osztály.Kulcsmező=érték
```

Ami ebből ismeretlen, az a Namespace (névtér), és itt el is kanyarodunk egy kicsit: a névtér arra való, hogy a különböző WMI osztályokat kategóriákba sorolhassuk. A CIMV2 (CIM version 2) névtér pl. a CIM szabvány által definiált osztályok és azok leszármaztatott gyermekosztályainak a tárolóhelye, a FrameworkV1 a .NET Framework WMI osztályainak, a directory nevű névtér az Active Directory osztályainak elkülönítésére szolgál. Kapcsolódáskor mindig csak egy azaz egy névtérhez kapcsolódunk, tehát normál esetben csak ennek a névtérnek az osztályait érhetjük el. Ha nem adtunk meg névteret, a root/CIMV2 névtérhez kapcsolódunk. A WMI teljes mértékben kiegészíthető új névterekkel és WMI providerekkel, azaz a kedves Olvasó is megírhatja saját WMI osztályát, amelyet majd beilleszt a WMI nagy motorjába, így a kedves Olvasó által írt szoftver is WMI-aware lesz, amely minden rendszergazda álma. ☺

Vissza a monikerekhez: a moniker a WMI programozó életét hivatott leegyszerűsíteni azzal, hogy különböző műveleteket egy sor kóddal megoldhasson. Így például a

```
WinMgmts:
```

moniker használatával egyszerűen a helyi WMI szerver CIMV2 (default) névteréhez csatlakozunk, míg a

```
WinMgmts://dszabotitanium/Root/CIMV2
```

alkalmazásával egy távoli WMI szerveréhez. A

```
WinMgmts://Win32_Share.Name='Tools'
```

moniker a Tools nevű hálózati megosztás WMI objektumát adja vissza.

Pár további érdekes példa:

Kissé „useless", de gyakorlásnak nem rossz: a következő VbScript példa a szerverünkre bejelentkezett felhasználók wallpaper képeinek elérési útvonalát dobja ki:

```
Dim objServices
Dim objUserDesktop

'Kapcsolódás a helyi WMI service-hez
Set objServices = GetObject("WinMgmts:")

'Végigiterálunk a bejelentkezett felhasználókon
For Each objUserDesktop In objServices.ExecQuery(
ᵇ "SELECT Name, Wallpaper FROM Win32_Desktop
ᵇ WHERE Wallpaper <> '(None)'")

    WScript.Echo objUserDesktop.Name & ": " &
    ᵇ objUserDesktop.Wallpaper

Next
```

A következő VbScript példa új IP címet kér a DHCP szervertől:

```
Dim objServices
Dim objNetworkAdapterConfig
Dim objNetworkAdapterConfigs

'Csatlakozás a helyi WMI szolgáltatáshoz
Set objServices = GetObject("WinMgmts:")

'Lekérdezzük az ethernet adapter konfigurációját
Set objNetworkAdapterConfigs =
ᵇ objServices.ExecQuery("SELECT * FROM
ᵇ Win32_NetworkAdapterConfiguration WHERE
ᵇ DhcpEnabled = TRUE and IpEnabled = True")

'Vesszük az első Dhcp-s IP címet
For Each objNetworkAdapterConfig In
ᵇ objNetworkAdapterConfigs
    Exit For
Next

'Eldobjuk az IP címet
MsgBox "Click to release IP address"
ᵇ objNetworkAdapterConfig.ReleaseDHCPLease

'Újat kérünk a DHCP szervertől
MsgBox "Click to renew IP address"
ᵇ objNetworkAdapterConfig.RenewDHCPLease
```

Jogosultság/Hitelesítés

A WMI névtér szintű jogosultság kezelést végez, ami azt jelenti, hogy névterenként különböző jogosultsági listákat (ACL-eket) állíthatunk be. Mindez a wmimgmt.msc MMC snap-inből elérhető.

A WMI alapértelmezésben NTLM/Kerberos hitelesítést alkalmaz. Ha ezt nem tudjuk vagy nem akarjuk használni, kapcsolódáskor megadhatunk más felhasználónevet és jelszót is. Fontos tudnunk, hogy ha a helyi WMI szerverhez kapcsolódunk, csak a futtató felhasználó nevében intézhetünk kéréseket (nem adhatunk meg más felhasználónevet/jelszót). Ez egy DCOM megszorítás.

A továbbiakban a teljes jogosultság/hitelesítés kezelése megegyezik a DCOM-éval: impersonation, access checks stb.

Hálózati elérés

Gyakori probléma, hogy a WMI kérések elszállnak a szigorú tűzfal/alapkonfiguráció beállításai miatt. Ne feledjük, hogy mind az RPC, mind a DCOM elég igényesek ebből a szempontból. Minden eshetőségre felkészülve, készítettem egy checklist-et, amelyet a kedves Olvasó felhasználhat mind WMI, mind DCOM vagy RPC hozzáférési problémáinak megoldására:

Elengedhetetlen beállítások:

- Activation and Launch jogosultság a kérést intéző felhasználónak a szerver DCOM konfigurációjában (dcomcnfg). A DCOM általános beállítások módosítása után minden esetben újra kell indítanunk a szerverünket!
- TCP port 135 (RPC root) nyitva a firewall-on
- RPC port intervallumok definíciója a registry-ben (részletesen: **[1]**), és ezen portok nyitott állapota a firewall-on (a netsh.exe lehetővé teszi port intervallumok definiálását Windows Firewall-on)
- Ha aszinkron WMI műveleteket végzünk, ugyanezen beállításoknak meg kell lenniük a kliensen is!

Hibaelhárítási lépések:

1. A script futtatása a szerveren megmutatja, hogy a probléma hálózati eredetű-e
2. Ping
3. A Windows firewall logja megmutatja, hogy mely TCP portra próbált csatlakozni a kliens sikertelenül (Control panel → Windows Firewall → Advanced → Security Logging)
4. Ha explicit felhasználónevet/jelszót használunk (ugyanazt, mint a logon user-ünk), könnyen leellenőrizhetjük, hogy a probléma Kerberos eredetű-e. Ha a probléma így nem jelentkezik, a Kerberos a hunyó.
5. Ha a probléma Kerberos eredetű, a klist.exe vagy kerbtray.exe eszközök megmutatják, hogy milyen ticketünk van (vagy nincs ☺) a WMI szerverre. Fontos, hogy ha a WMI szerverünket újraindítjuk hibaelhárítás közben, a boot-olás után akár 1-2 percig is eltarthat, amíg Kerberos kapcsolatot tudunk létesíteni vele. Eddig a pillanatig viszont minden kérésünk sikertelen lesz. Az előbb említett eszközök arra is jók, hogy a cache-elt Kerberos jegyeinket kidobjuk.

Dokumentáció

Jó hír, hogy a WMI nagyon jól dokumentált. A kedves Olvasó leellenőrizheti, ha a **[2]** oldalon beírja bármely WMI osztály nevét, az eredménylista első eleme mindig az adott osztály dokumentációja lesz. A megfelelő WMI osztály keresésére viszont a WMI Administrative Tools ingyenesen letölthető **[3]** csomag használatát javaslom. Feltelepítése után a WMI CIM Studio browser-ben települő ActiveX controllal bármilyen WMI szerverhez csatlakozva böngészhetjük a teljes osztályhierarchiát, osztályokat kereshetünk név, vagy akár dokumentáció alapján, lekérdezéseket vagy listázásokat futtathatunk.

Tipp: mindig fordítsunk külön figyelmet az alkalmazásunk (ha lehet mindegyik, de legalább a WMI ☺ részének) különböző operációs rendszeren történő tesztelésére, ne felejtsük el, hogy a WMI egy Windows API felett álló réteget alkot, amely nagyban függ az alatta lévő réteg kompatibilitásától.

Összefoglalás

A WMI használatával nagyban leegyszerűsíthetjük alkalmazásaink menedzsmentjét. Nagy előnye, hogy nem kell beleásnunk magunkat különböző technológiákba vagy protokollokba, egy egyszerűen programozható felületen keresztül elérjük bármilyen WMI kompatibilis alkalmazás management funkcionalitását. A Windows minden management funkciója elérhető WMI-on keresztül (software, hardware) is, ezen kívül a Microsoft termékek nagy része és más gyártók különböző szoftverei is kellemes meglepetéseket okozhatnak projektünk utolsó napjain.

Szabó Dávid
dszabo@microsoft.com
terméktámogató mérnök
MCSD

A cikkben szereplő URL-ek:

[1] http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dndcom/html/msdn_dcomfirewall.asp
[2] http://msdn.microsoft.com/library
[3] http://www.microsoft.com/downloads

Az ADAM címtár használata I.

ADAM belülről

Nagyjából másfél évvel ezelőtt jelent meg a TechNet Magazin oldalain az ADAM címtárról szóló első cikkem, amelyben általánosságban mutattam be az akkor még újdonságnak számító szoftvert. Azóta ADAM kicsit idősebb lett (bár új verzió nem jelent meg), született néhány olyan alkalmazás, amely ezt a címtármegoldást is használja, itt az ideje, hogy részletesebben is megismerkedjünk vele.

A cikk első részében áttekintjük ADAM legfontosabb tulajdonságait, majd feltelepítünk egy ADAM példányt, a vele érkező felügyeleti eszközök segítségével rövid felderítőútra indulunk ADAM belső szervei közt, és végrehajtunk néhány kisebb műtétet is. A második részben pedig a .NET keretrendszer eszközeit vesszük elő, programból végzünk el néhány olyan műveletet, amelyet Active Directory (különösen esetleg éles rendszer) esetében nem szívesen próbálnék ki: új alkalmazás-partíciót hozunk létre, bővítjük a sémát, és néhány ezer (esetleg néhány tízezer) objektum létrehozásával próbára tesszük a címtár sebességét és skálázhatóságát.

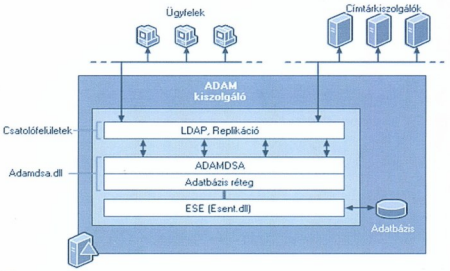
Ki is ez az ADAM?

Az Active Directory Application Mode (ADAM) az Active Directory speciális üzemmódja, amely a címtárat használó alkalmazások fejlesztésekor számos különféle helyzetben előnyösen használható. Az ADAM nemcsak a tartományvezérlőkön, hanem a tag kiszolgálókon, vagy önálló gépeken is futtatható, sőt több példányban is elindítható, és a példányok mindegyikéhez önálló beállításokat adhatunk meg. Sok alkalmazásnak csak egészen egyszerű címtárra van szüksége. A tárolt információkat általában csak szűk körben kell elérni, és nincs szükség a teljes vállalatra kiterjedő replikációra sem. Az alkalmazások kiszolgálása más szolgáltatási módot igényel, mint amit a Windows hálózati infrastruktúra kezelésére szolgáló Active Directory nyújtani tud.

Mi a teendő például, ha egy alkalmazás, vagy portál igényei miatt néhány felhasználóról nyilván kell tartanunk olyan adatokat is, amelyek nem szerepelnek az AD-ben? Ha az AD sémát bővítjük, az egész erdőben replikálódni fognak a változtatások, egy nagyobb vállalat esetében ez hosszadalmas, bonyolult, és erőforrás-pazarló megoldás. Ha viszont létrehozunk egy ADAM példányt, amely az extra tulajdonságokat tárolja, az alkalmazás a felhasználók hitelesítésére továbbra is használhatja az AD-t (ez sok szempontból kellemesebb megoldás, mint az ADAM alapú hitelesítés), az extra információkat pedig az ADAM példányhoz fordulva érheti el.

ADAM használatával az alkalmazások címtárai folyamatosan fejlődhetnek, változhat a séma és a replikációs beállítások, hogy a fejlesztés alatt álló alkalmazással együtt növekedhessen a hozzá tartozó címtár is. Az ADAM példányok külön-külön módosíthatók, nincs szükség egyetlen alkalmazás miatt a teljes vállalat címtár-struktúrájának megváltoztatására.

Az ADAM a címtárat megvalósító szolgáltatásból (ennek kódbázisa megegyezik az Active Directory-val), a hozzá tartozó adatbázisfájlból, valamint a hozzáférést biztosító csatolófelületekből áll.



Az ADAM felépítése

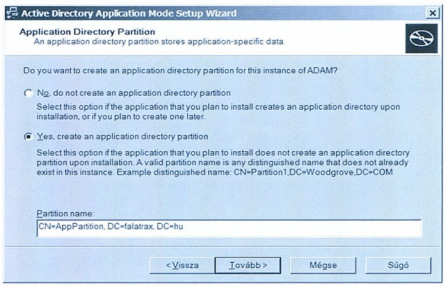
Replikáció

Az ADAM az Active Directoryval megegyező multi-master replikációs modell szerint működik, így az adatok a replikációban részt vevő bármelyik adatbázisban módosíthatók. Az ADAM hasonlóan az Active Directory-hoz lehetővé teszi a telephelyek közötti replikáció ütemezését és az átvitt adatok tömörítését is. Ha az alkalmazások adatait az Active Directoryban tároljuk, azok replikációja csak a teljes adatbázissal együtt történhet. Az ADAM használatával az alkalmazások

adatainak replikációja azok speciális igényeinek megfelelően is ütemezhető. Természetesen lehetőség van akár egy gépen belül az ADAM példányok közötti replikációra is.

Telepítés

Az ADAM a Windows Server 2003 kiegészítő komponense, a Microsoft külön csomagban terjeszti azt. A csomag letölthető az **[1]** címről. Négy letölthető fájl közül választhatunk, mivel a szoftver létezik x86 és IA64 platformra, valamint van végfelhasználói (retail) és viszonteladói (redistributable) változat is, amelynek licence lehetővé teszi, hogy az általunk készített alkalmazásokhoz csomagolva továbbadhassuk azt. Az exe fájlt kibontva, a telepítést az adamsetup.exe futtatásával indíthatjuk el. A telepítő első képernyőjén azt kell kiválasztanunk, hogy az ADAM-ot és felügyeleti eszközeit, vagy csak a felügyeleti eszközöket szeretnénk telepíteni. A felügyeleti eszközök természetesen képesek távoli gépeken futó ADAM példányokhoz való csatlakozásra is. Ezután meg kell határoznunk, hogy egyedi példányt (unique instance), vagy már meglévő példány replikáját (replica of an existing instance) szeretnénk telepíteni. A második opció használatával könnyen megoldható az ADAM példányok másolása, ha például arra van szükség, hogy a nagy munkával beállított, egyedi sémával felszerelt példányunkat másik gépre telepítsük át. A telepítő ebben az esetben minden sémaváltozást átvisz az új példányra is. Ezután meg kell adnunk az új ADAM példány nevét, és azokat a portokat, amelyeken keresztül elérhető lesz. Alapértelmezés szerint az ADAM a 389-es és a 636-os (SSL esetén) portot figyeli, de természetesen minden egyes példány számára külön portszámokat kell megadnunk. A telepítő utolsó kérdése arra vonatkozik, hogy szeretnénk-e alkalmazás partíciót létrehozni a telepítés során:



Alkalmazás partíciót is létrehozunk

ADAM az adatokat hierarchikus felépítésű címtár-fájlban tárolja, amelynek alapértelmezett helye:

```
Program Files\Microsoft ADAM\<példány neve>
↳ \Data\adamntds.dit
```

A tároló logikai partíciókra, vagy másik terminológia szerint névterekre (naming contexts) van felosztva. Három partíciótípust különböztetünk meg: konfigurációs, séma és alkalmazás partíciókat. Minden címtárban kötelezően van egy, és csakis egy konfigurációs és séma partíció, így ezek az ADAM telepítésekor automatikusan létrejönnek. Alkalmazáspartícióból viszont több is lehet, ezek közül egyet létrehozhatunk a telepítés során, de később is gyárthatunk újakat, amint azt ki is fogjuk próbálni az Ldp program, és .NET komponensek segítségével is.

A három partíciótípus szerepe a következő:

- Konfigurációs partíció – ez a partíció az ADAM példány beállításait tartalmazza, ide kerülnek például a replikációval kapcsolatos paraméterek, a biztonsági beállítások, stb.
- Séma partíció – itt tárolódnak a címtárban definiált objektum és attribútum típusok. A séma partícióban nemcsak a „gyári" elemeket találhatjuk meg, hanem ide kerülnek azok az objektum típusok és attribútumok is, amelyekkel mi magunk bővítjük a sémát. Ha a címtárban új objektumot szeretnénk létrehozni, a definíciónak (osztálynak) szerepelnie kell a séma partícióban.
- Alkalmazás partíció – ebben a partícióban tárolódnak az ADAM címtárat használó alkalmazások egyedi információi. Külön partíciót hozhatunk létre minden kapcsolódó alkalmazáshoz, vagy egyetlen alkalmazás különböző igényeinek kiszolgálására. Az alkalmazás partícióban olyan tárolókat (containers), és objektumokat hozhatunk létre, amelyek típusa, és attribútumai a séma partícióban szerepelnek. Minden alkalmazáspartícióra önálló tárolási, terjesztési és replikációs beállítások adhatók meg.

Minden partíciónak (és minden más címtárobjektumnak is) van egy megkülönböztető neve (distinguished name, DN), ezt kell megadnunk a létrehozáskor, és ezzel hivatkozhatunk rá később is. Ha új partíciót szeretnénk létrehozni (akár a telepítőből, akár más módszerrel), ezt a nevet kell megadnunk. A név több részből áll, amelyek mindegyike a hierarchikus struktúra egy csomópontját azonosítja, hasonlóan a fájlrendszerbeli útvonalakhoz. Minden rész egy DN attribútumból és a hozzá tartozó értékből áll. Az attribútumok a következők lehetnek:

- DC – tartománynév összetevő (Domain Component)
- C – ország (Country)
- L – hely (Location)
- O – szervezet (Organization)
- OU – szervezeti egység (Organizational Unit)
- CN – objektumnév (Common Name)

Hozzunk létre a telepítőben egy alkalmazás partíciót, amelynek megkülönböztető neve:

```
CN=AppPartition, DC=falatrax, DC=hu
```

A következő képernyőn megadhatjuk a címtárfájl tárolómappáját, majd ki kell választanunk azt a felhasználói fiókot, amelynek nevében az ADAM példányt megvalósító rendszerszolgáltatás fog futni. Minden példány külön Windows szolgáltatást hoz létre, így természetesen a futtató felhasználó is különböző lehet valamennyi példány esetében. Ezután megadhatjuk azt a felhasználói fiókot, vagy csoportot, amelynek joga lesz az adott példány felügyeleti műveleteinek elvégzésére. A telepítés utolsó lépéseként kiválaszthatjuk az ADAM telepítőcsomagban érkező LDIF fájlok közül azokat, amelyeket már most szeretnénk a címtárba importálni (később a parancssori ldifde.exe program használatával végezhetjük el ugyanezt a műveletet). Az LDIF fájlok ebben az esetben „csak" a sémát bővítik, de mivel az LDIF a címtárak közötti replikáció szabványos formátuma, segítségükkel bármilyen más művelet is elvégezhető. Importáljuk az „MS-User.ldf" fájlt, amely új objektum (person, user, stb.) és attribútum típusokat (title, user-comment, picture, stb.) ad hozzá a séma partícióhoz.

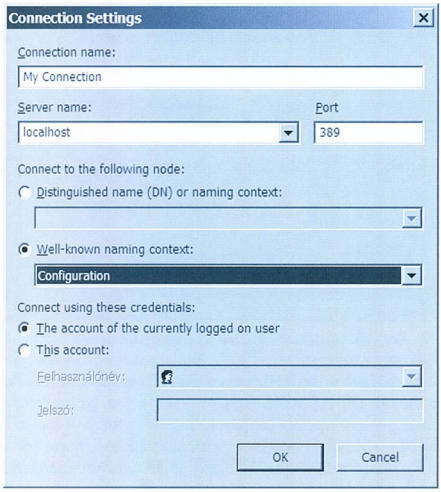
Ha a telepítő sikeresen lefutott, az ADAM felügyeleti eszközeit a `%SYSTEMROOT%\ADAM` mappában találhatjuk meg, a rendszerszolgáltatások közé pedig bekerül az ADAM_<példány neve> néven futó szolgáltatás, amely rendszerindításkor automatikusan elindul.
Ha újabb példányt telepítünk, a telepítőprogram már másik portot ajánl fel az új példány eléréséhez (alapértelmezés szerint 50001, és 50002), és az egyes példányok eltávolítását is egymástól függetlenül végezhetjük el.

Az ADAM felügyeleti eszközei

Mivel az ADAM az Active Directory egy üzemmódjának tekinthető, a felügyelet az ott megismert eszközök módosított változataival végezhető el. Az ADAM felügyeleti eszközei az alkalmazással együtt kerülnek telepítésre:

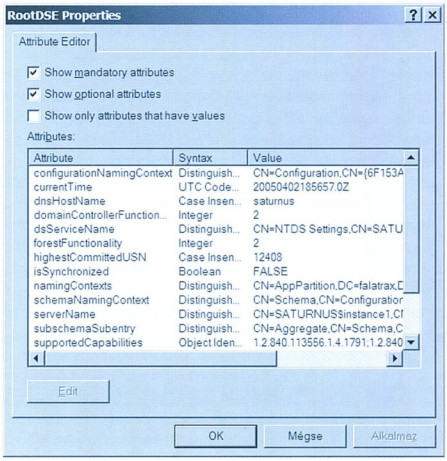
- **Ldp** (Ldp.exe) – az ldp segítségével (ami igen egyszerű, de grafikus felhasználói felülettel működik) LDAP műveleteket végezhetünk az ADAM címtáron.
- **Ldifde** (ldifde.exe) – a program segítségével végezhető el a speciális formátumú LDIF fájlok címtárba importálása, és a címtáradatok fájlba exportálása is. (A telepítőprogram is ezt az eszközt használta az „MS-User.ldf" fájl importálásához.)
- Az **ADAM Schema** mmc modul segítségével megjeleníthetjük és módosíthatjuk az ADAM címtár sémáját.
- Az **ADAM ADSI Edit** (ADAM-adsiedit.msc) az ismert ADSI Edit eszközön alapul, és lehetővé teszi a címtár valamennyi objektumának (a sémának és a beállítási adatoknak is) megjelenítését, az objektumok módosítását és a hozzáférés vezérlési listák szerkesztését.

Indítsuk el elsőként az ADAM ADSI Edit modult, és vizsgáljuk meg a címtár felépítését. Kattintsunk jobb gombbal az ADAM ADSI Edit sorra, és válasszuk a Connect to... parancsot. Adjuk meg a megfelelő adatokat, és a „Well-known naming contexts" listában válasszuk ki a Configuration sort.



Csatlakozunk a Configuration partícióhoz

Láthatjuk, hogy a Configuration névtér legfelső konténerének neve: CN=Configuration, CN={GUID}. Ebben találhatjuk meg a CN=Partitions konténert, amely tartalmazza a címtár valamennyi partíciójának DN-jét.
Ha csatlakozunk a Schema partícióhoz is, láthatjuk, hogy az egyetlen konténerből áll. amelynek neve CN=Schema, CN=Configuration, CN={GUID}. Ebben találhatók a címtárban létrehozható objektumok osztály és attribútum típusai. Ezek egyik része, a címtár alapértelmezett készleteihez tartozik, míg másokat mi magunk hoztunk létre, amikor a telepítés során beimportáltuk az „MS-User.ldf" fájlt. Ha egy kicsit előre gondolkodunk, máris kezdhetünk aggódni, hogyan is fogunk programból hozzáférni ezekhez a partíciókhoz, ha nem tudjuk „fejből" a DN-jükben szereplő GUID értéket. Szerencsére létezik egy harmadik névtér is, az úgynevezett RootDSE, amely akár bejelentkezés nélkül is minden odatévedő program számára olvasható, és amelynek tulajdonságai között megtalálhatjuk a keresett információt. A „configurationNamingContext", és a „schemaNamingContext" tulajdonságok tartalmazzák a megfelelő partíciók DN-jeit. Sőt még ennél is többet megtudhatunk, mivel a „namingContexts" tulajdonság a címtár valamennyi alkalmazáspartíciójának DN-jét is tartalmazza (a configuration és schema partíciókét is), vagyis ha kiolvassuk ezeket az információkat, akár még teljesen ismeretlen felépítésű címtár alkalmazás-partícióira is rá tudunk akaszkodni. Csatlakozzunk a harmadik „well-known" névtérre (RootDSE), amely egyetlen konténerből áll, ennek tulajdonságlapja látható az alábbi képen, itt találhatjuk meg az előbb említett tulajdonságok értékeit:

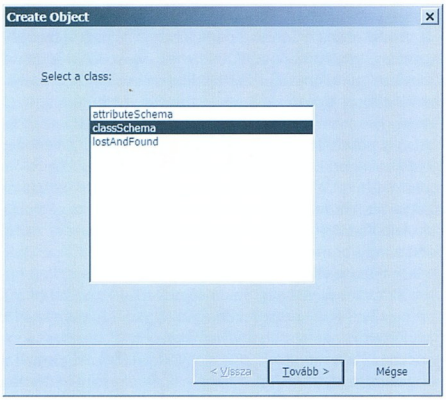


A RootDSE konténer tulajdonságlapja

Sémabővítés

Most pedig nézzük, hogyan bővíthetjük saját objektum és attribútum típusokkal az ADAM sémát, és hogyan hozhatunk létre ezeknek a típusoknak megfelelő objektumot az alkalmazás-partícióban. Indítsuk el az ADAM-ADSI Edit konzolt, és ha még nem tettük meg csatlakozzunk a séma partícióhoz. Jobb gombos kattintás után válasszuk a helyi menüből az Új -> Ob-

ject parancsot. A megjelenő párbeszédablakban láthatjuk, hogy milyen objektumtípusok létrehozására van lehetőség:

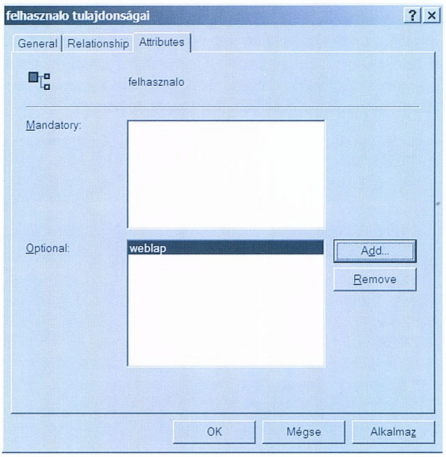


Új séma osztályt hozunk létre

Az itt megjelenő elemeket az adott konténer allowedChildClasses tulajdonsága tartalmazza. Válasszuk a classSchema elemet, és menjünk tovább. A következő ablakban az új objektum CN-jét kell megadnunk (CN= előtag nélkül), ez legyen mondjuk „felhasznalo". A subClassOf tulajdonság következik, most a user osztály módosított változatát szeretnénk létrehozni, így örökítsünk ebből (ezzel osztályunk hosszú öröklési lánc végére kerül: top -> Person -> organizationalPerson -> User). Általános használatra a top osztálytól célszerű örökíteni, ő az osztályhierarchia csúcsa. A user osztályt az „MS-User.ldf" fájl importálásával hozta létre a telepítőprogram, a mi új osztályunk ennek valamennyi saját és örökölt tulajdonságával rendelkezni fog, sőt egy új elemmel is bővítjük majd a készletet.
Az utolsó lapon a governsID tulajdonságot kell megadnunk, ez az objektum típus egyedi azonosítója. Ha szélesebb körben használt alkalmazás számára végezzük a sémabővítést, akkor a **[2]** weblapon találhatunk információt arról, hogyan regisztráltathatjuk séma osztályaink egyedi azonosítóját a Microsoftnál. Most azonban ennek semmi jelentősége, ha vaktában lövöldözünk, akkor sem túl valószínű, hogy már létező azonosítót sikerülne találni (ha mégis, válasszunk másikat). Aki szeret sokat gépelni, az használja mondjuk az 1.2.840.113556.1.6.1.2.1 értéket. Ezzel majdnem készen is vagyunk, de mivel az új objektumok és értékek első körben csak az ADSI gyorsítótárba kerülnek, értesítenünk kell a címtárat is a változásokról mielőtt használatba vesszük az új objektum típust. Kattintsunk jobb gombbal a séma partícióhoz létrehozott kapcsolatra, és válasszuk az „Update schema now" parancsot. Frissítsük a séma konténer tartalmát megjelenítő listát is (F5), ezután már biztosan megtaláljuk benne az újonnan létrehozott típusunkat.
Folytassuk egy attribútum típus létrehozásával, ezt majd hozzá fogjuk csatolni az objektum típusunkhoz. A kezdés ugyanaz, majd a megjelenő ablakokban rendre adjuk meg a következő értékeket:

- Class – AttributeSchema
- cn – weblap
- oMSyntax – 64 (ez unicode sztring típust jelent)
- IDAPDisplayName – weblap
- isSingleValued – true (az attribútum csak egyetlen értéket tartalmazhat majd)
- attributeSyntax – 2.5.5.12 (sztring)
- attributeID – 1.2.840.113556.1.6.1.1.1

Ha az attribútumban esetleg nem sztringet szeretnénk tárolni, az oMSyntax és attributeSyntax tulajdonságok megfelelő értékeiről a **[2]** címen kaphatunk felvilágosítást.
A cache-ben lévő adatok elküldése és frissítés után a listában megjelenik az új attribútum típus.
Már csak egy dolog van hátra: az objektum típusunkkal tudatni kell, hogy az attribútum típus az övé, használja egészséggel. Mi sem könnyebb ennél, megkeressük az objektumunkat, annak tulajdonságlapján az allowedAttributes mezőt, és hozzáfűzzük a listához az új értéket. Innen már csak néhány OK gombra kell kattintanunk, hogy megkapjuk a „Felépített attribútum módosítása nem engedélyezett" feliratú hibaüzenetünket. Sajnos ezt a műveletet NEM lehet az ADAM ADSI Edit konzolból elvégezni, kénytelenek leszünk az ADAM Schema nevű modult elővenni. Indítsunk el egy üres mmc –t, és adjuk hozzá az ADAM Schema modult. Kattintsunk jobb gombbal az ADAM Schema csomópontra, és válasszuk a Change ADAM Server... parancsot. Meg kell adnunk az ADAM kiszolgálót (localhost), a megfelelő portszámot (389), és a csatlakozó felhasználó nevét (bejelentkezett felhasználó). A Classes csomópont alatt keressük meg a típusunkat, jelenítsük meg a tulajdonságlapját, és az Optional mezőhöz adjuk hozzá az új attribútum típust.



Az új objektum típus új attribútuma

Természetesen, ha az új típus alapján objektumot hozunk létre, annak nem csak egyetlen attribútuma lesz, hiszen örökli a user, person, top, stb. osztályok valamennyi attribútumát is. A képen is látható, hogy csak opcionális attribútumokat adhatunk hozzá a típushoz, kötelező attribútumok megadására csak az osztály létrehozása közben van lehetőség (ha az osztályt a Schema modulban készítjük el). Ez a korlátozás kön-

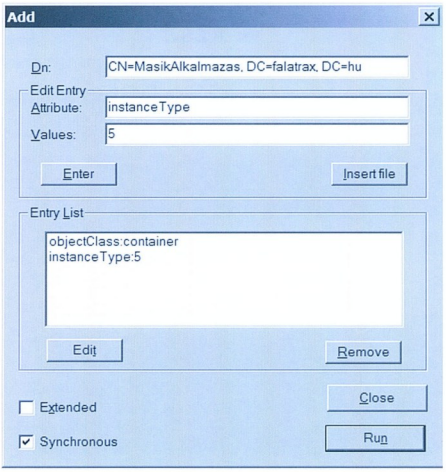
nyen érthető, ha meggondoljuk, micsoda meglepetés érné az osztály alapján már létrehozott objektumokat egy új kötelező attribútum megjelenésekor. Ha a tulajdonságlapot becsukjuk, a jobb oldali panelen látható a típus valamennyi (öröklött és saját) attribútuma, köztük a sajátunk is. Ha ezután visszatérünk az ADSI Edit konzolba, szomorúan tapasztalhatjuk, hogy az allowedAttributes listában továbbra sem jelenik meg az attribútumunk. Ha viszont az osztályra alapuló objektumot hozunk létre, annak weblap tulajdonsága már gond nélkül használható, amint azt a következő bekezdésben ki is fogjuk próbálni. Mielőtt nagyon belemelegednénk a sémabővítésekbe, jó, ha tudjuk, hogy az ADAM (a legtöbb más címtárhoz hasonlóan) nem teszi lehetővé a létrehozott osztályok, vagy attribútum típusok törlését. Típusainkat azonban „üzemen kívül" helyezhetjük, ha az isDefunct tulajdonságot igazra állítjuk. A példányhoz tartozó Windows szolgáltatás újraindítása után már nem hozhatunk létre az adott típusra alapuló objektumokat.
A már létező objektumok természetesen megmaradnak, objectClass tulajdonságukban már nem az osztály neve, hanem governsID-je fog szerepelni. Célszerű tehát a tervezett bővítéseket alaposan átgondolni, és egy olyan mintapéldányon letesztelni, amelyet a kísérletek után nyugodtan letörölhetünk.

Objektum létrehozása az alkalmazás partícióban

Az új objektum létrehozásához ismét az ADAM ADSI Edit konzolt fogjuk használni. Először is csatlakozzunk a telepítéskor létrehozott alkalmazás partícióhoz. Ez természetesen már nem „well-known" névtér, így meg kell adnunk a „CN=AppPartition, DC=falatrax, DC=hu" megkülönböztető nevet. Keressük meg a névtér legfelső konténerét, és jobbklikk után válasszuk az Új -> Object parancsot. Ha minden jól sikerült (és az ADSI cache-t is elküldtük már a címtárba) akkor a létrehozott „felhasznalo" típusnak meg kell jelennie a listában. Hogy egy adott típus megjelenik-e itt, azt a típus possSuperior tulajdonságának értéke határozza meg. Itt tételesen fel kell sorolnunk azokat a típusokat, amelyek tartalmazhatják őt. Alapértelmezés szerint ide bekerül a „container" típus, ez most nekünk éppen elég is. Ha más típusú tárolókban (például szervezetben, vagy szervezeti egységben) is szeretnénk létrehozni az osztályunkon alapuló objektumot, akkor ezeket is fel kell vennünk a tulajdonság értékei közé. (Legjobb, ha ezt ismét a Schema modulban tesszük, mégpedig az adott osztály tulajdonságlapjának „Relationship" oldalán.)
Válasszuk tehát a „felhasznalo" típust. Már csak egyetlen tulajdonságot kell megadnunk (CN=SziklaSzilard), és készen is vagyunk. Az új objektum tulajdonságlapján már megjelenik a „weblap" tulajdonság, az értéket is itt állíthatjuk be.

Új alkalmazás partíció hozzáadása

Végül próbáljunk ki még egy lehetőséget; hozzunk létre új alkalmazás partíciót címtárunkban. Ehhez újabb eszközt fogunk használni, mégpedig az ADAM csomaggal együtt érkező ldp.exe programot. Indítás után csatlakozzunk a címtárhoz (Connection -> Connect), itt meg kell adnunk a kiszolgáló nevét (localhost), és a csatlakozáshoz használandó portot (389). A Connection -> Bind paranccsal a bejelentkezést végezhetjük el, ha egyetlen mezőt sem töltünk ki, az aktuális felhasználó próbál bejelentkezni. A Browse -> Add Child parancsra kattintva a következő párbeszédablakot kapjuk (persze kitöltetlenül:):



Új alkalmazás partíció létrehozása

Konténer objektumot hozunk létre, ennek négy kötelező attribútuma van (cn, objectClass, instanceType, objectCategory), de ha az utolsót nem adjuk meg, az automatikusan generálódik az objektum létrehozásakor. A másik három értéket írjuk be az ábrának megfelelően, ezután a Run gombra kattintva létre is jön az új partíció.

A cikk következő részében már a .NET keretrendszer osztályait felhasználva fogjuk többé-kevésbé ugyanezeket a műveleteket elvégezni, és próbára tesszük az ADAM címtár sebességét is.

Szerényi László
szerenyi.l@met.hu

A cikkben szereplő URL-ek:

[1] http://www.microsoft.com/windowsserver2003/adam/default.mspx
[2] http://msdn.microsoft.com/library/en-us/ad/ad/obtaining_an_object_identifier.asp
[3] http://msdn.microsoft.com/library/en-us/adschema/adschema/syntaxes.asp

ASP.NET 2.0 (Whidbey)

Mi várható a 2005. évi ASP.NET-ben?

IV. RÉSZ

Az ASP.NET Cache és az SQL Server 2005 együttműködése

Bevezetés

Az előző részben megnéztük az ASP.NET Cache alapvető használatát, illetve adatbázisfüggő Cache-elés felépítését SQL Server 2000 háttérrel. Ebben a részben megvizsgáljuk, milyen új Cache vezérlési lehetőségeket aknázhatunk ki az SQL Server 2005 nyújtotta új szolgáltatások felhasználásával.

Adatbázisfüggő Cache felépítés alapjai

Emlékeztetőül a probléma leírása. Adatbázisból lekérdezések eredményhalmazát jelenítjük meg weblapokon. A lekérdezések erőforrásigényesek, ezért szeretnénk minimalizálni őket. A lekérdezések adatait tárolhatjuk a futó weblapok kódjával azonos AppDomainben, az ASP.NET Cache objektumban. Azonban az adatbázisadatok véletlenszerű időzítéssel változhatnak, amely után a gyorsítótárba helyezett adatokat ki kell üríteni, illetve a következő kérésnél újra kell építeni. A fő probléma: hogyan értesüljön róla a tár, hogy a lekérdezett adatok változtak? Kinek is van ismerete elsőkézből a változásokról? Természetesen az adatbázisnak. Ha az adatbázis vissza tudna szólni a Cache-nek, hogy az adatok változtak, akkor a problémát megoldottuk.

Ehhez két követelményt kell teljesíteni:

1. Az adatbázis tudjon róla, hogy egy bizonyos lekérdezés, - amely akár WHERE feltételt, JOIN-t, stb. is tartalmaz – kimenete megváltozott egy adatmódosító művelet hatására. Azaz az adatbázisnak meg kell jegyezni a kitüntetett lekérdezéseket, és figyelni az adatmódosításokat, melyek érintenék a korábbi adatokat. Ez nem egyszerű probléma.
2. Az adatbázis képes legyen aszinkron módon értesíteni az ASP.NET Cache-t a megváltozott adatokról. Az aszinkron követelmény miatt kiesett a „triggerrel közvetlenül beszólok a Cache-nek" megoldás, mert a szinkron Cache kezelés nagyon lelassítaná az adatbázismódosításokat.

A második követelmény miatt szükséges valamiféle átmeneti tároló, ahol várakoznak a Cache értesítések feldolgozásra. Ha igazságos rendszert akarunk létrehozni, aki elsőnek jött, azt kell elsőként kiszolgálni, azaz egy várakozási sorra, egy Queue-ra lesz szükség. Ezt a sort már lehetne triggerrel táplálni, ám triggerrel még mindig nehéz lenne az 1. követelményt implementálni. Szerencsére az SQL Server 2005-ben mindkét követelményre van megoldás. A lekérdezés eredményhalmazának változását a Query Notifications nevű technológia képes figyelni, az események aszinkron továbbítását egy várakozási soron keresztül pedig a Service Broker nevű új SQL Server komponens képes elvégezni.

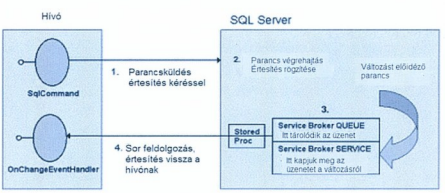
Bár a cikk az ASP.NET Cache-ről szól, nagy vonalakban érdemes megismerni ezen két szereplőt, így jobban megértjük az SqlCacheDependency működését is.

SQL Server 2005 Query Notifications

A szolgáltatás építőkövei a következők:

- SQL Server 2005 Query Engine
- SQL Server Service Broker
- sp_DispatcherProc (CLR) tárolt eljárás
- SqlNotification osztály (System.Data.Sql.SqlNotificationRequest)
- SqlDependency osztály (System.Data.SqlClient.SqlDependency)

Az ADO.NET SqlCommand kapott egy új jellemzőt a 2.0-ban, a Notification-t. Ha a parancs futtatása előtt ezt kitöltjük egy SqlNotificationRequest példánnyal, akkor ezzel jelezzük az SQL Servernek, hogy kérünk értesítést, ha a futtatott lekérdezés eredményhalmaza megváltozik. Az SQL Server ezek után „figyel" minden adatmódosítást, hátha az érinti a korábban regisztrált értesítendők egyikét. Ha változás történt, akkor a Query Engine elküld egy üzenetet a Service Broker várakozási sorába. Ezek után az értesítés vagy visszamegy közvetlenül a hívóhoz, vagy a hívó kérdezi le van-e a számára értesítés. A folyamatot szemlélteti a következő ábra:



Az SQL Server 2005 Query Notifications működése

A következő kérdések az izgalmasak:

1. Honnan tudja az SQL Server, hogy egy lekérdezés eredményhalmaza megváltozott?
2. Hogyan szól vissza a hívónak?

Az eredményhalmaz változásfigyelés valójában már benne volt az SQL Server 2000-ben is. Amikor ugyanis indexelt né-

zetet hoztunk létre, akkor az SQL Servernek tudnia kellett átvezetni a nézet alapját adó táblák változásait a letárolt nézetre. Mivel ugyanez a mechanizmus működteti az értesítéseket, ezért ugyanazok a kötöttségek vonatkoznak a felhasznált lekérdezésekre. Pl. UNIONt, DISTICTet, OUTER JOINt, allekérdezést, és még jó pár rázósabb parancsot nem lehetett használni indexelt nézetekben, és ezekre az értesítések se mennek. Miért? Mert nagyon nehéz megmondani, hogy egy tábla változás hatására változik-e egy, a fenti bűnös parancsokat használó lekérdezés eredményhalmaza. Eme háttérismeret azért nagyon fontos, mert az indexelt nézetek dokumentációja alapján máris tudhatjuk, hogy a Query Notifications fog-e működni a konkrét lekérdezésünkre. Ami „elég egyszerű", arra menni fog.

Nézzük a második kérdést. Mi történik, ha a szerver észreveszi, hogy értesítést kell küldenie? A Query Engine a Service Broker egy SERVICE-éhez küldi az értesítést.

A Service Broker egy új elem az SQL Server 2005-ben, a 2000-nek még nem volt része. Ő tulajdonképpen egy adatbázisban megvalósított várakozási sor, amelyben SERVICE-ek létesítenek kommunikációs végpontokat. A SERVICE egy olyan végpont, amely adattárolását egy QUEUE végzi el, és egy CONTRACT (gyakran XML Schema) írja le a SERVICE-ben feldolgozható üzenetek formátumát. Azaz egy QUEUE-ban lehet több SERVICE, és minden SERVICE-hez csatolhatunk CONTRACT-okat.

A Query Notifications CONTRACT-ja (szerződése, formátumleírása) be van építve az SQL Serverbe, és a következő URL azonosítja:

```
http://schemas.microsoft.com/SQL/Notifications/
fPostQueryNotification
```

Ez csak egy azonosító, nem tölt be semmit az SQL Server a fenti URL-ről. Kifinomult esetben létrehozhatunk saját SERVICE-t, saját QUEUE-ban is, de az alap értesítések kedvéért már léteznek ezek az msdb adatbázisban. Miután az értesítések elkezdenek szaporodni az alapértelmezett QUEUE-ban, ki lehet őket olvasni (pollozással). De azt ígértem, hogy az SQL Server képes visszaszólni a hívónak! És tényleg. A már említett sp_DispatcherProc hozzá van rendelve az alapértelmezett Query Notification SERVICE-hez, az dolgozza fel, ha értesítések esnek be. Ő egy .NET-ben írt tárolt eljárás, amely felolvassa a várakozási sorba érkezett értesítéseket, és saját protokoll segítségével visszaszól az ügyfélnek. A saját protokoll lehet TCP és HTTP, de ez nem azonos az SQL parancsok végrehajtására használt, TDS csomagokat szállító csatornával, különösen, hogy ez a kiszolgáló felől nyílik meg az ügyfél irányába. Mivel ez egy független csatorna, az értesítések kedvéért nem kell nyitva tartani a normál adatbázis-kapcsolatot, az értesítések „hátulról", független szálakról érkeznek be a hívóba. Amikor az ügyfél megkapta az értesítést, az sp_DispatcherProc törli a QUEUE-ból az értesítési kérést, és indítja a következő feldolgozást.

Az SQL Server 2005 Query Notifications programozása – SqlDependency

Habár belülről igen összetett az értesítő architektúra, programozni meglepően egyszerű. Az SqlDependency osztály egységbe zár minden részletet, ami az értesítések kezeléséhez szükséges. Csak annyi a teendőnk, hogy hozzárendeljünk egy SqlDependency példányt a futtatandó SqlCommandhoz, és rákapcsolódjuk az OnChanged eseményére. Ügyelni kell arra, hogy a parancs olyan legyen, amiről az SQL Server 2005 képes értesítést küldeni, ezt kitárgyaltuk az előző fejezetben. A példánk egyszerű SELECT lesz, de ebben is kéttagú táblaneveket kell használni, hogy a tulajdonos (SQL Server 2005-ben schema) egyértelmű legyen. E nélkül nem kapnánk értesítést.

Lássunk hát egy működő vázat:

```
using System;
using System.Data;
using System.Data.SqlClient;

class App {
  static void Main() {
    string connString = "Data Source=.;Initial
↳ Catalog=AdventureWorks;Integrated
↳ Security=true;";
    using (SqlConnection conn =
      new SqlConnection(connString))
    // 2-tagú táblanevek kellenek,
    //mint az indexelt nézetekben
    using (SqlCommand cmd =
      new SqlCommand(
      "SELECT Description, StartDate FROM
↳ Sales.SpecialOffer", conn)) {
      try {
        //Az értesítést kezelő objektum
        // hozzárendelése a parancshoz
        SqlDependency depend =
          new SqlDependency(cmd);
        //Az eseménykezelő regisztrációja
        depend.OnChanged +=
          new  OnChangedEventHandler(OnChanged);

        conn.Open();
        SqlDataReader rdr = cmd.ExecuteReader();
        //Eredményhalmaz feldolgozása
        while (rdr.Read())
          Console.WriteLine(rdr[0]);
        rdr.Close();

        //Várunk az értesítésekre
        Console.WriteLine("ENTER");
        Console.ReadLine();
      }
      catch (Exception e) {
        Console.WriteLine(e.Message);
      }
    }
  }

  static void OnChanged(object caller,
    SqlNotificationEventArgs e) {
    Console.WriteLine("A lekérdezés
↳ eredményhalmaza megváltozott.");
    Console.WriteLine("Source " + e.Source);
    Console.WriteLine("Type " + e.Type);
    Console.WriteLine("Info " + e.Info);
  }
}
```

Az SqlDependency konstruktorban lehet átviteli csatornát választani, hitelesítést kérni, stb. Ezeket most nem tárgyalom részletesen, főleg, mert ezek még változhatnak a végleges változatig.

Kapcsolat az ASP.NET Cache-sel

Ha már ilyen szépen kidolgozták az adatváltozásokról szóló értesítéseket, nem volt túl nehéz alkalmazniuk azt ASP.NET környezetre is. Az ASP.NET Cache eleve rendelkezik függőségkezeléssel, így külső esemény vagy változás hatására képes kiütni valamilyen tárolt tartalmat. Ami új a 2,0-ban, hogy kapunk adatbázis-függőségi kezelést is, az SqlCacheDependency személyében.

Ez az osztály kétféleképpen tud működni: az előző részben bemutatott SQL 7 és 2000-re épített pollozós módon, ezt láttuk, hasznos, de nem túl izgalmas.

SQL Server 2005 esetén már sokkal izgibb a játék, hisz a Query Notification nem bután a tábla változásaira reagál, hanem szelektíven egy adott lekérdezésre hangoltan működik. Nézzünk egy példát az SqlCacheDependency kézi használatára, az előző részben látott mintára építve:

```
public partial class SqlCacheDep :
  System.Web.UI.Page
{
  protected void Page_Load(object sender,
    EventArgs e)
  {
    DataSet cachedData = (DataSet)Cache["d"];
    if (cachedData == null)
    {
      SqlCacheDependency dependency;
      cachedData =
        LoadFromDatabase(out dependency);
      dg.DataSource = cachedData;
      Cache.Insert("d", cachedData, dependency);
    }
    else
    {
      dg.DataSource = cachedData;
    }
    dg.DataBind();
  }

  private DataSet LoadFromDatabase(
    out SqlCacheDependency dependency)
  {
    using (SqlConnection conn =
      new SqlConnection("..."))
    {
      SqlDataAdapter adapter =
        new SqlDataAdapter(
        "SELECT Description, StartDate FROM
Sales.SpecialOffer", conn);
      dependency = new
        SqlCacheDependency(adapter.SelectCommand);
      DataSet ds = new DataSet();
      adapter.Fill(ds, "SpecialOffer");
      return ds;
    }
  }
}
```

A lap azonban hibával száll el:

```
System.Data.SqlClient.SqlException: User "NT
AUTHORITY\NETWORK SERVICE" does not have
permission to request query notification
subscriptions on database "AdventureWorks".
A severe error occurred on the current command.
The results, if any, should be discarded.
```

Ez aztán az irgum-burgum! Viszont logikusan jött ez a hiba, mert az értesítések adatbázis erőforrásokat kötnek le, ezért védeni kell őket. Adjunk hát jogot a weblapunkat futtató felhasználónak:

```
GRANT SUBSCRIBE QUERY NOTIFICATIONS TO
[NT AUTHORITY\NETWORK SERVICE]
```

Erre kapunk egy másik hibát:

```
Cannot find the object
"QueryNotificationErrorsQueue" because it does not
exist or you do not have permissions
```

A QUEUE-hoz RECEIVE jogra van szükségünk, hogy ki tudjuk venni belőle az értesítést:

```
GRANT RECEIVE
ON dbo.[QueryNotificationErrorsQueue]
TO [NT AUTHORITY\NETWORK SERVICE]
```

Persze sokkal egyszerűbb lett volna, ha beraktam volna a webfelhasználómat a sysadmin szerepkörbe, és akkor minden simán menne. Amint láttuk, némi pluszmunkával ugyan, de gyenge felhasználóval, minimális jogosultságokkal is lehet sikereket elérni, ráadásul a rendszerünk sokkal védettebb lesz a támadásokkal szemben.

Az OutputCache együttműködése az SQL Server 2005-tel

Az OutputCache-t nagyon szeretjük, mert egyszerű használni, és az egész lap html kimenetét letárolja, ezért általában sokkal hatékonyabb, mint az előbbi DataSet alapú gyorsítás. Ha szeretnénk az OutputCache-t függővé tenni az SQL 2005 Query Notifications-től, akkor ezt a fejlécet kell berakni a lap elejére:

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="OutputCacheDep.aspx.cs"
Inherits="OutputCacheDemo" %>
<%@OutputCache SqlDependency="CommandNotification"
Duration="86400" VaryByParam="none"%>

<html xmlns="http://www.w3.org/1999/xhtml" >
...
  <body>
    <form id="form1" runat="server">
        <asp:Label id="time" runat="server"/><p/>
        <asp:DataGrid id="dg"
         AutoGenerateColumns="true"
         runat="server"/>
    </form>
  </body>
</html>
```

Az SqlDependency="CommandNotification" jelöli ki, hogy nem pollozásra, hanem valódi értesítésre vágyunk.

A lapot mozgató kód rendkívül egyszerű, simán kiolvassuk az adatokat DataAdapter vagy DataReader segítségével:

```
public partial class OutputCacheDemo:
  System.Web.UI.Page
{
  protected void Page_Load(...)
  {
    time.Text = DateTime.Now.ToString();
    dg.DataSource = LoadFromDatabase();
    dg.DataBind();
  }

  private DataSet LoadFromDatabase()
  {
    using (SqlConnection conn =
      new SqlConnection("..."))
    {
      SqlDataAdapter adapter =
        new SqlDataAdapter(
        "SELECT Description, StartDate FROM
↳ Sales.SpecialOffer", conn);
      DataSet ds = new DataSet();
      adapter.Fill(ds, "SpecialOffer");
      return ds;
    }
  }
}
```

Az ASP.NET automatikusan hozzákapcsolja az SqlCacheDependency-t a végrehajtott parancshoz, és az majd mindent elintéz a háttérben!
A lap teszteléséhez egyszer le kell futtatni az aspx oldalt, utána egy napig nem fog változni a lap tetején látható idő, hacsak meg nem változtatjuk az adatokat, pl. így:

```
UPDATE Sales.SpecialOffer
SET Description = Description + '1'
WHERE SpecialOfferID = 1
```

Játsszunk kicsit az értesítő architektúrával!

```
UPDATE Sales.SpecialOffer
SET Description = Description + '1'
WHERE 0 = 1
```

Triviális trükk lett volna, természetesen nem kapunk értesítést.

```
UPDATE Sales.SpecialOffer
SET Description = Description
WHERE SpecialOfferID = 1
```

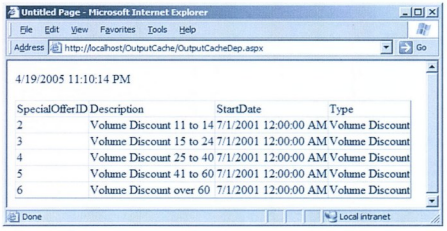
Ennek bedől, és üríti az OutputCache-t. Tanulság: ne tessék kuka, valójában semmit nem csináló UPDATE-eket írni. Végül is a triggerek is lefutnak az ilyen parancsra, a notification is buzgón jelez. Mert ő ilyen. Inkább szól kisebb megmozdulásokra is, semmint a Cache tartalma valami elavult vacakot tartalmazzon.

```
ALTER TABLE Sales.SpecialOffer
ADD UjOszlop INT NULL
```

Ez ugyan nem módosít adatokat a táblában, de azért ez elég nagy volumenű módosítás, hogy kapjunk róla hírt. Kísérletezzünk az értesítések szelektivitásával is! Írjuk át a lekérdezést így:

```
SELECT SpecialOfferID, Description, StartDate,
[Type] FROM Sales.SpecialOffer
WHERE [Type] = N'Volume Discount'
```

Kimenet:



Lőjünk mellé egy UPDATE-tel:

```
UPDATE Sales.SpecialOffer
SET Description = Description
WHERE SpecialOfferID = 1
```

Mivel ez nem érinti az előbb leválogatott sorokat, a Cache háborítatlan marad! Ez nagyon ügyes, ezt már nem tudta az SQL Server 2000-re épülő megoldás.

Soczó Zsolt
zsolt.soczo@netacademia.net
A szerző a NetAcademia vezető fejlesztőoktatója
ASP.NET MVP, MCSE, MCSD, MCDBA, MCT

A cikkben szereplő URL-ek:
[1] netacademia.net/tudastar/articlepage.aspx?upid=5876

Windows szolgáltatások 7. rész

LocalSystem a mi várunk

Újra tekintélyes mennyiségű Windows Server 2003 szolgáltatás működését, jellemzőit ismertetjük meg sorozatunkban.

Print Spooler
(Nyomtatásisor-kezelő)

A szerviz rövid neve: Spooler
Az alkalmazás neve: spoolsv.exe
Függés: Remote Procedure Call
Függesztés: Fax, Print Server for Macintosh, TCP/IP Print Server
Porthasználat: TCP: 139, 515; UDP: 137, 138, 1709
Alapértelmezett indítás: automatikus

Valószínűleg mindenki által jól ismert ez a szolgáltatás, hiszen az összes helyi illetve hálózati nyomtatási sor kezelőjéről van szó. A nyomtatási alrendszer egy olyan központi eleme ez a szerviz, amelyen biztosan „átmegy" az összes nyomtatási feladat. A Print Spooler kommunikál a printer meghajtókkal valamint az I/O komponensekkel (pl. az USB porttal vagy a TCP/IP-vel). A nyomtatási feladatok további kezelése, azaz egyrészt a megfelelő printer driver megkeresése majd betöltése, illetve a nyomtatási feladatok sorbaállítása és időzítése is e szerviz feladata.

Az indítása automatikus, és folyamatosan a számítógép leállításáig működik is. Ha leállítjuk vagy letiltjuk, akkor mind a nyomtatás, mind a faxolás lehetősége megszűnik, akkor is, ha nem helyi nyomtatóra/faxra nyomtatunk. Valamint - mivel nem igény szerint indul/áll le - nem észleli az induló nyomtatási feladatot, ergo csak akkor tilthatjuk le, ha a szerveren nincs beállított helyi nyomtató/fax illetve a szerverünk egy-egy hálózati nyomtató nyomtatási sorát sem kezeli. Ezzel a szervizzel kapcsolatban meg két dologra hívnám fel a figyelmet. Az egyik az ingyenesen letölthető Windows Server 2003 Resource Kit Tools-ban **[1]** szereplő Print Spooler Information (SplInfo.exe), amely egy parancssori segédeszköz a nyomtatási sorral kapcsolatos információk összegyűjtésére és képernyőre listázására (akár egy távoli számítógépről is). A másik pedig egy csoportházirend beállítás, amely segítségével könnyedén engedélyezhetjük/tilthatjuk a kliensek hozzáférését.

```
Computer Configuration\Administrative
Templates\Printers\Allow Print Spooler to accept
client connections
```

Ha ez az opció nincs beállítva, akkor a spooler addig nem fogad el kéréseket, amíg nincs egy megosztott helyi nyomtató, vagy egy hálózati nyomtató nyomtatási sora nincs beállítva. Ha az opció engedélyezve van, akkor minden kérést elfogad, ha viszont le van tiltva, semmilyet. A változtatáshoz a szervizt újra kell indítani.

Protected Storage
(Védett tároló)

A szerviz rövid neve: ProtectedStorage
Az alkalmazás neve: lsass.exe
Függés: Remote Procedure Call
Függesztés: -
Porthasználat: -
Alapértelmezett indítás: automatikus

Ez a szerviz az érzékeny adatainkat - úgymint jelszavak, privát kulcsok - védi, a nem jogosult felhasználóktól vagy éppen szolgáltatásoktól, processzek-től. A különböző alkalmazások (Outlook, IE, stb.) számára megengedi, hogy egyszerűen elhelyezzék és szükség esetén kinyerjék a lerakott kódokat, jelszavakat. Erre a célra egy biztonságos és jól elzárt tárolót használ, amely sérthetetlenségét a felhasználó mesterkulcsán (master key - a profilba rejtett, a felhasználó jelszavából képzett, ám többszörösen „megkevert", háromhavonta megváltozó kulcs) keresztül a HMAC (Hash-Based Message Authentication Code) módszerrel és az SHA1 algoritmus segítségével biztosítja a DPAPI (Data Protection API). Ez egy teljes szélességében beavatkozás nélküli folyamat, nem is konfigurálható.

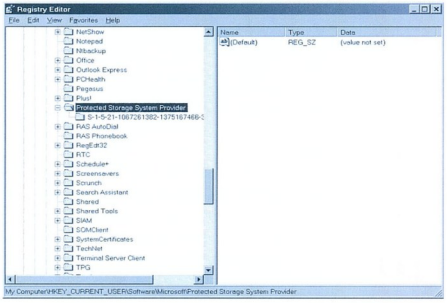
Ha letiltjuk vagy leállítjuk a szervizt, akkor az ilyetén védett adatok elérhetetlenek lesznek, a tanúsítványkiadó szolgáltatás nem működik tovább, sőt az S/MIME és az SSL sem, valamint az intelligens kártyás (smartcard) belépés is lehetetlenné válik. Van ezzel a szervizzel kapcsolatban még egy érdekesség, nemcsak a leállásakor/letiltásakor lehetnek fennakadások, hanem egy speciális esetben is. Ez úgy derült ki, hogy miután az IE-nek megengedtük, hogy tárolja a különböző weboldalakon rendszeresen használt és ezért kényelmi szempontból elmentett jelszavakat, és közben az Outlooknak is, hogy a POP3 postafiókok jelszavát jegyezze meg, egyszercsak mindkettővel elkezdődtek a problémák. Az Outlook megtagadta a belépést a postafiókba, míg a jó ismerős weblapoknál állandóan egy „Protected Storage Service" panel akarta kikényszeríteni az elvileg már ismert jelszavakat. A megoldáshoz vezető úton (ami persze a Google, mint mindig) kiderült, hogy a valószínűleg az adott profilban generált jel-

szó cache és a Védett tároló szervizéhez kapcsolódó regisztrációs adatbázisban tárolt tartalom ütközik (mert mondjuk megsérült az utóbbi). Mit lehet tenni? Találtunk megoldást:

- le kell állítani a szervizt
- el kell indítani regedt32.exe-t és ide navigálni:

```
HKEY_CURRENT_USER \ Software \ Microsoft \
Protected Storage System Provider
```

- ahogy a képen is látható egyetlen kulcs található itt, az adott felhasználó SID-jével. Nos, ezt kellett törölni. Persze (és anno ezért kellett a regedt32, viszont ma már nem kell, jó a sima regedit.exe is), előtte adtunk magunknak jogosultságot (jobb gomb a kulcson > Permissions...), mert alapból csak a SYSTEM felhasználónak van hozzáférése.



A SID-del jelölt kulcsot kell törölni

- ezután indítsuk el a szervizt és a hiba megszűnik.

Figyelem! A megoldás egyetlen szépséghibája, hogy a javítás mellett egyúttal fergeteges rombolást és sóval beszórást végez a korábban itt-ott elmentett jelszavak között, de ezeket újra megadva (az alkalmazásokban is) működni fog minden szépen.

Remote Access Auto Connection Manager
(Távelérés – automatikus kapcsolódás kezelője)

A szerviz rövid neve: RasAuto
Az alkalmazás neve: rasauto.dll (svchost.exe)
Függés: Remote Procedure Call, Remote Access Connection Manager, Telephony, Plug and Play
Függesztés: -
Porthasználat: -
Alapértelmezett indítás: kézi, leállítva

Ezt a szolgáltatást másképp „Autodial service"-nek is nevezik, és ebből már könnyebb kitalálni, hogy egy automatikus kapcsolódást próbál megvalósítani, akkor, amikor egy alkalmazás egy távoli DNS vagy NetBIOS nevet szeretne elérni. A szerviz először biztosra akar menni, azaz megpróbálja feloldani a távoli számítógép vagy pl. egy megosztás nevét, vagy hálózati csomagokat próbál küldeni a távoli gép felé, mert élő hálózati kapcsolat esetén természetesen nem lesz automata „tárcsázás". Ám ha ezek a módszerek eredménytelennek bizonyulnak, akkor aktiválódik, és feldob egy modemes vagy VPN csatlakozási lehetőséget a felhasználó elé. Nyilván csak a már meglévő kapcsolatokat képes ilyenkor felajánlani a csatlakozáshoz, ezek közül pedig azt, amelyet legutóbb használtunk (mert az összerendeléseket tárolja) az adott hely eléréséhez.

Ha leállítjuk, kézzel kell indítanunk a csatlakozást, amennyiben szükséges. De van, ami még ekkor is elindul! Ez pedig a Windows termékaktiváció, amely a Telefonos szerviz mellett ezt is képes beizzítani szükség esetén. Ha viszont letiltjuk a szervizt, akkor nem, és természetesen más alkalmazás sem.

Remote Access Connection Manager
(Távelérési csatlakozáskezelő)

A szerviz rövid neve: RasMan
Az alkalmazás neve: rasmans.dll (svchost.exe)
Függés: Remote Procedure Call, Plug and Play, Telephony
Függesztés: Internet Connection Firewall (ICF) / Internet Connection Sharing (ICS), Remote Access Auto Connection Manager
Porthasználat: -
Alapértelmezett indítás: kézi, leállítva

E szerviz feladata a telefonos és a VPN kapcsolatok kézben tartása. Amikor a Hálózati kapcsolatok ablakban duplán kattintunk egy kapcsolatra, akkor ez a szerviz tárcsáz, illetve küldi a VPN kapcsolódási kérést és ezután is kezeli a kapcsolatot, pl. egyeztet a RAS szerverrel.

Ha nincs működésben egy-egy kapcsolat, akkor a szerviz távozik a RAM-ból, persze, ha megnyitjuk a Hálózati kapcsolatok ablakot, akkor megint betöltődik, hiszen a kapcsolatok állapotának lekéréséhez is szükség van rá. Abban az esetben, ha nincs egyetlen táveléréshez szükséges kapcsolatunk, akkor viszont nem kell, marad alapállapotban. Ha letiltjuk, vagy leállítjuk, nem tudunk modemes vagy VPN kapcsolatot létrehozni, és az esetleges bejövő kapcsolatok sem fognak működni. Vizuálisan is szegényebbek leszünk, mert a Hálózati kapcsolat ablak sem mutat tartalmat, még a helyi kapcsolatokat sem.

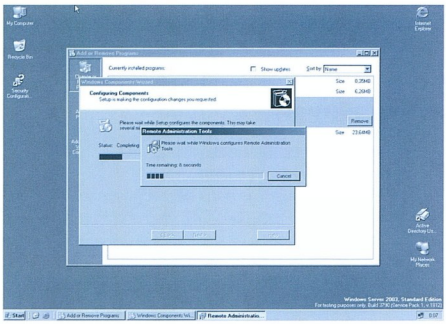
Remote Administration Service
(Távfelügyeleti szolgáltatás)

A szerviz rövid neve: SrvcSurg
Az alkalmazás neve: Srvcsurg.exe
Függés: Remote Procedure Call
Függesztés: -
Porthasználat: -
Alapértelmezett indítás: automatikus

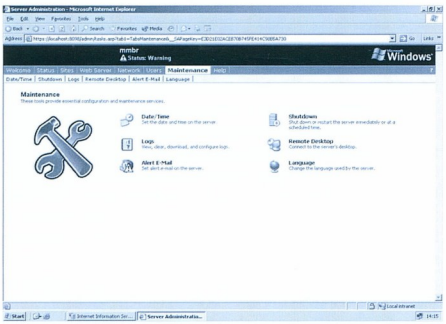
Ez a szolgáltatás kifejezetten érdekes. Az összes szervizekkel kapcsolatos hivatalos dokumentum szerint az operációs rendszer (bármelyik Windows 2003 változat) alapértelmezett telepítésével kerül fel a rendszerre, ám akárhány Windows 2003 szervert tekintettem meg (legalább tízet), egyszer sem láttam a szolgáltatások listájában, de a futtatható állományt sem a lemezen. Ha viszont feltelepítem az IIS-t (ami ugye, nem default már) és vele a „Remote Administration (HTML)" komponenst, akkor feltelepül az 5 MB-os „Remote Administration Tools" csomag is, meglesz az .exe is és a szerviz is. Érdekes...

Mindenesetre maga a Remote Administration szolgáltatás a következő, rendszerindításkor lezajló műveleteket hajtja végre:

- Eggyel növeli a szerverindítások számának értékét
- Generál egy self-signed tanúsítványt (így aztán akkor is tud https-t használni, ha mi nem telepítettünk)
- Egy figyelmeztetést kreál, ha a dátum és/vagy az időbeállítás nem korrekt
- Egy másik figyelmeztetést kreál, ha az e-mail küldés nem konfigurált vagy nem is lehetséges



A Remote Administration Tools csomag külön települ



És ez az „eredmény" - https://localhost:8098

A szolgáltatás akkor indul, ha a Remote Server Manager a COM interfészen keresztül meghívja. A COM interfész csak a LocalSystem fiók és az Administrators csoport tagjai számára tudja megengedni ezt a kérést a kliensről, ami egy biztató dolog. Ám vegyük figyelembe, hogy az ilyen típusú távirányítás biztonsági szempontból nem egy kívánatos módszer mondjuk egy nyilvános hálózaton keresztül. Ezért célszerű vagy letiltani a szerviz indítást, vagy fel sem telepíteni a csomagot, vagy legalább csoport/helyi házirenden keresztül szabályozni a szervizhez való hozzáférést, és aztán kézzel az indítást/leállítást.

Remote Desktop Help Session Manager
(Távoli asztal súgó-munkamenetének kezelője)

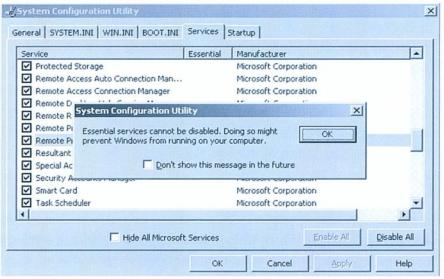
A szerviz rövid neve: RDSessMgr
Az alkalmazás neve: sessmgr.exe
Függés: Remote Procedure Call
Függesztés: -
Porthasználat: -
Alapértelmezett indítás: kézi, leállítva

Egyszerű szerviz, a Remote Assistance (Távsegítség) szolgáltatás Súgón belüli (ugyanis ha elakad a felhasználó, a Súgóból megkérheti a rendszergazdát, hogy kapcsolódjon már a gépéhez) használatát ellenőrzi és kezeli.
Ha leállítjuk vagy letiltjuk, azon kívül, hogy a feladatát nem teljesíti, semmilyen más hátrány nem ér bennünket, így ha nem kell, bátran tegyük csak meg.

Remote Procedure Call
(Távoli eljáráshívás)

A szerviz rövid neve: RpcSs
Az alkalmazás neve: rpcss.dll (svchost.exe)
Függés: -
Függesztés: ***[2]***
Porthasználat: TCP: 135, 530, 593, 80, RPC over HTTP esetén dinamikus; UDP: 135, 530, 2034 ill. dinamikus
Alapértelmezett indítás: automatikus

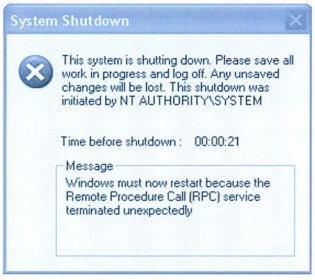
Minden szervizek atyja, egy alap Windows Server 2003 rendszerben kb. 53 szerviz függ tőle. Ezt a listát nem részletezem, viszont a letölthető, 311 oldalas *„Threats and Countermeasures: Security Settings in Windows Server 2003 and Windows XP"* dokumentumban megtekinthető [2]. A távoli eljáráshívás (RPC) a Windows operációs rendszerek által ősidők által használt szerver/kliens protokoll. Az Open Software Foundation távoli eljáráshívási protokollon alapul, amelyet a Microsoft kibővített néhány saját elemmel. Az RPC segítségével egy alkalmazás egyszerűen hozzáférhet egy másik számítógépen futó szolgáltatáshoz egy IPC (InterProcess Communication = egymástól független folyamatok közti adatcsere) csatornán keresztül. Ezek a folyamatok lehetnek ugyanazon a gépen, vagy egy hálózat gépein, vagy akár két, az internetre csatlakozó gépen is.



Az MSCONFIG tiltakozik

Ezt a szervizt Windows Server 2003 esetén semmilyen jogosultsággal nem sikerült leállítani vagy letiltani, sem a Services MMC-ből sem az MSCONFIG segédprogramból.

Ha történetesen valahogy mégiscsak sikerülne leállítani, vagy az indulási értéket letiltani (ez Windows 2000-nél lehetséges és az XP-nél is, közvetve, a DCOM Server Process Launcher szolgáltatáson keresztül), akkor az operációs rendszer védekezésképpen újraindul, és ha letiltottuk, akkor az újraindítás után számtalan, igazán komoly problémával szembesülünk, szinte semmi nem működik, ha egyáltalán beindul a Windows. Ami biztos: ezek után elindítani sem fogjuk tudni ezt a szervizt, ergo marad a Recovery Consol, amivel viszont újra automatikusan induló állapotba tudjuk tenni. Szóval csak nagyon óvatosan.



A Blasternek anno sikerült lelőni az RPC szervizt

Remote Server Manager
(A távoli kiszolgáló kezelője)

A szerviz rövid neve: AppMgr
Az alkalmazás neve: Appmgr.exe
Függés: Remote Procedure Call
Függesztés: -
Porthasználat: -
Alapértelmezett indítás: automatikus

A Remote Server Manager szoros együttműködésben van a korábban említett Remote Administration szervizzel, és a következőket nyújtja:

- tárolja a Remote Administration figyelmeztetéseket,
- megmutatja, törli és számolja is ezeket,
- a Remote Administration feladatok végrehajtását segíti.

Ha a szervizt kézi indításúra tesszük, akkor egy Remote Administration feladat vagy egy figyelmeztetés miatt el fog indulni elindulni. Ha letiltjuk, nem lesznek figyelmeztetések. Az indításával és az engedélyezésével kapcsolatban a problémák ugyanazok mint a Remote Administration szolgáltatásnál.

Removable Storage
(Cserélhető tároló)

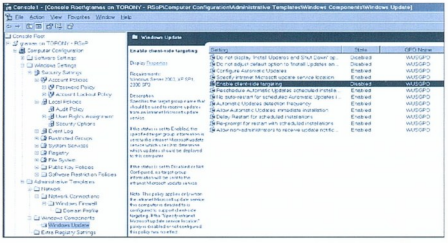
A szerviz rövid neve: NtmsSvc
Az alkalmazás neve: ntmsvc.dll (svchost.exe)
Függés: Remote Procedure Call
Függesztés:
Porthasználat: -
Alapértelmezett indítás: kézi, leállítva

Az eltávolítható médiák meghajtóit és tárolóit kezeli. A szalagos meghajtók, a CD/DVD-k vagy akár a ZIP meghajtók katalógusait információt gyűjti és rendszerezi, valamint hozzáférést nyújt (mount/unmount/eject) az eltávolítható médiákhoz (rendszerint az NTBackup és a Remote Storage szerviz használja). A Removable Storage által kezelt eszközöket megtekinthetjük a Computer Management MMC-ben a Storage szakaszban. Ha leállítjuk vagy letiltjuk, ezeket a feladatokat természetesen nem fogja teljesíteni. Igazából letiltani nem sok értelme van, hiszen a szerviz amúgyis csak akkor indul el (automatikusan), ha egy erre épülő alkalmazás használni szeretné és csak addig működik, amíg használatban van. Ha viszont mégis letiltjuk, a tapasztalatom szerint az NTBackup valami egészen elképesztően lassan indul el és habozik később is.

Resultant Set of Policy Provider
(Az eredő házirend szolgáltatója)

A szerviz rövid neve: RSoPProv
Az alkalmazás neve: RSoPProv.exe
Függés: Remote Procedure Call
Függesztés: -
Porthasználat: -
Alapértelmezett indítás: kézi, leállítva

Mint ahogyan a nevéből is kiderül, ez a szerviz az RSoP kiszolgálója. Lehetővé teszi, hogy csatlakozzunk a tartományhoz, elérjük a távoli WMI adatbázist és az RSoP segítségével szimuláljuk a csoportházirend beállításait. A szervizt elsősorban az RSoP MMC használja, és persze az RSoP-Planning Mode Provider/WMI Provider összes csatlakozó kliense is. Az RSoP a csoportházirend egy remek kiterjesztése, amellyel egyrészt egy MMC modulból kockázat nélkül tesztelhetjük a tervezett csoportházirend beállítások hasznát adott felhasználókra, vagy számítógépekre (tervező üzemmód), egyúttal hibakeresésre is használhatjuk, illetve a beállításokat is ellenőrizhetjük, jelentések formájában (naplózó üzemmód). Gyakorlatilag az RSoP egy lekérdező motor, amely az ún. CIMOM (Common Information Management Object Modell) adatbázisból, a WMI segítségével összegyűjti (telephely, tartomány, DC, vagy szervezeti egység szinten) a jelenlegi és a tervezett beállításokat. A CIMOM adatbázis független a címtártól, viszont minden a hálózatba belépett gép automatikusan feltölt egy sor magára jellemző adatot ide, pl. a hardverről, az IE-ről, a mappaátirányításokról, a biztonsági beállításokról, stb..



Az RSoP naplózó üzemmódban

Az RSoP másik nagy előnye, hogy ha több szinten (site, domain, OU), több különböző csoportházirend objektum van beállítva, akkor ezek hatása (a beállításoktól és a szabályoktól függően) ütközhet, ám az RSoP képes eligazodni ebben, és mindig az aktuálisan helyen hatályos és érvényes beállítást mutatja meg.
Ha a szervizt leállítjuk vagy letiltjuk, az RSoP tervező mód elérhetetlen lesz az adott DC-n. Más ismert hatása nincs.

Routing and Remote Access
(Útválasztás és távelérés)

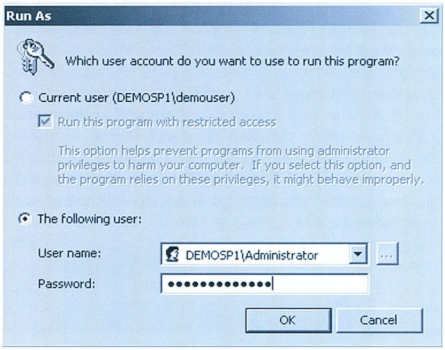
A szerviz rövid neve: RemoteAccess
Az alkalmazás neve: mprdim.dll (svchost.exe)
Függés: NetBIOS Interface, NetBIOS Group, Remote Procedure Call
Függesztés: -
Porthasználat: TCP: 47 (GRE) 1723; UDP: 500, 1701 (Layer 2 Tunneling) 1723 (PPTP)
Alapértelmezett indítás: letiltva

Ki ne ismerné az RRAS-t? És aki ismeri, tudja, hogy rengeteg mindenre használható, valódi komplex megoldás. Telefonos/modemes kiszolgáló, teljes körű VPN szolgáltatás, LAN-LAN, LAN-WAN útválasztás, NAT, mini DHCP/DNS kiszolgáló, házirendek, igény szerinti tárcsázás – mind-mind megtalálhatóak benne. Rengeteg komoly könyv és cikk is íródott már erről a témáról, pl. ebben az újságban is volt már egy remek sorozat **[3]**, így aztán nem szaporítanám feleslegesen a szót. Annyit azonban megjegyeznék, hogy amennyiben az RRAS MMC-ben engedélyezzük a szolgáltatást és együttal elindítjuk az RRAS varázslót, akkor a szerviz a letiltott állapotból automatikusan indulóra vált át. Viszont ez nem jelenti azt, hogy ha ezek után bármikor leállítjuk a Services MMC-ben, akkor szintén „igényre" el fog indulni. Azaz ilyenkor a szokásos „piros nyíl karikában" látványban lesz részünk. Ha a konzolról tiltjuk le, akkor viszont a szerviz is letiltásra kerül és értelemszerűen a fenti feladatokat nem fogja ellátni.

Secondary Logon
(Másodlagos bejelentkezés)

A szerviz rövid neve: Seclogon
Az alkalmazás neve: seclogon.dll (svchost.exe)
Függés: -
Függesztés: -
Porthasználat: -
Alapértelmezett indítás: automatikus

A Windows 2000 esetén „*RunAs Service*" néven futott ez a szolgáltatás, valószínű, hogy sokan ezen a nevén is ismerik. Nagyon hasznos szerviz, különösen akkor, ha erőt véve magunkon, komoly elszánással elhatározzuk, hogy nem az Administrators csoport tagjaként fogunk dolgozni a jövőben a gépünkön. Ez nemes és respektálható cél, de azért néha mégis szükséges, hogy legyen magasabb jogosultságunk. Kilépés/belépés helyett viszont használhatjuk ezt a szolgáltatást arra, hogy az adott alkalmazást, processzt egy másik felhasználó jogosultsági környezetében futtassuk, természetesen csak addig, amíg be nem zárjuk az alkalmazást. Ritkán fordított szituációban is megtörténhet a felhasználása: szeretnénk gyorsan megtudni, hogy egy adott felhasználó képes lesz-e az adott programot futtatni, eléri-e a szükséges mappákat, állományokat a saját jogosultsági szintjével.



A kép magáért beszél

Másik előnye ennek a szerviznek, a parancssori RunAs.exe működése. Ezzel alkalmazásokat, MMC konzolokat és egyéb parancsikonokat is képesek vagyunk „izmosan" futtatni:

```
runas /user:domain_nev\user_nev mmc.exe
```

Vagy akár közvetlenül a Disk Management bővítményt:

```
runas /user:domain_nev\user_nev diskmgmt.msc
```

Amennyiben egy Control Panel parancsikont (.cpl) szeretnénk futtatni, használjuk ezt a listát **[4]**. Tartományi Windows 2000 Professional esetén nekem csak a „control" szóval együtt ment:

```
runas /user:administrator "control appwiz.cpl"
```

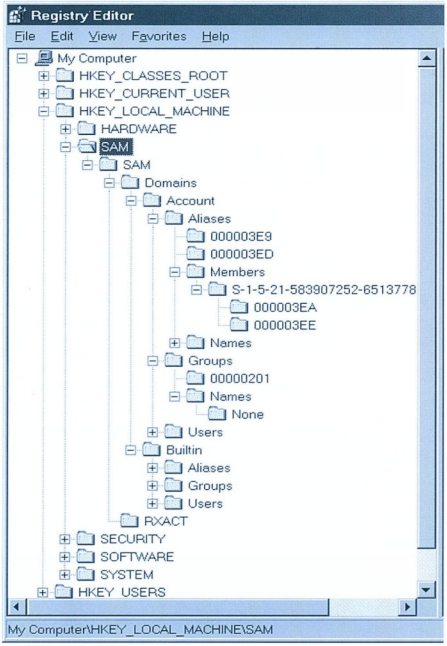
További tippekhez Aaron Margosis blogját **[5]** ajánlanám, ahol viszonylag kevés, de annál remekebb cikket olvashatunk ebben a témában.
Ha a szervizt leállítjuk, szükség esetén elindul, ha letiltjuk, nem használhatjuk sem a parancssori eszközt, sem a helyi menüben lévő megfelelőjét.

Security Accounts Manager
(Biztonsági fiókkezelő)

A szerviz rövid neve: SamSs
Az alkalmazás neve: lsass.exe
Függés: Remote Procedure Call
Függesztés: DHCP Server, Distributed File System, Distributed Transaction Coordinator, IIS Admin Service, FTP Publishing Service, HTTP SSL, World Wide Web Publishing Service, Microsoft POP3 Service, Network News Transfer Protocol, Simple Mail Transfer Protocol, Intersite Messenger Service, Windows Internet Name Services, Message Queuing, Message Queuing Downlevel Client Support, Message Queuing Triggers
Porthasználat: -
Alapértelmezett indítás: automatikus

A SAM (Security Accounts Manager) egy olyan kiemelt biztonsági komponens, amely az LSA-val, a NetLogon szerviszel, LSA Server-rel és az SSP/AP-vel (Security Support Provi-

ders/Authentication Package) együtt az lsass.exe processz biztonsági környezetében fut. Gyakorlatilag ezek egy Windows operációs rendszer biztonsági sarokkövei. A SAM feladata az, hogy a (helyi) felhasználók, csoportok, és jelszavak (Windows 2000-től kezdve a munkaállomás fiókok is) tárolását intézze. A SAM a Samsrv.dll-ben „lakik" a tárolandókkal együtt, az említett adatbázis pedig önmagában a regisztrációs adatbázis HKLM\SAM ágában is megtalálható (plusz a \System32\Config\SAM állományban). Az említett HKLM\SAM ágat hiába keressük, csak a SYSTEM felhasználó láthatja, azaz még az Administrator sem.



Amit elvileg nem lehet látni

Ezen adatok tárolása egy tartományba léptetett számítógépnél már az AD dolga, ám a helyi adatok ekkor is megmaradnak. A tartományvezérlő esetén azonban nem, hiszen ott nincs helyi adatbázis, pontosabban összesen egy felhasználó marad a SAM-ban, ez pedig a Directory Services Restore Mode fiók (amelyet az AD telepítésekor adunk meg, ergo meg nem egyenlő a helyi Administrator jelszavával!).

A szervizt természetesen nem lehet leállítani a Services MMC-ből. Le lehet viszont tiltani az újraindítás utáni automatikus indulást, de ne tegyük meg ezt a lépést semmiképp, mert számtalan komponens függ tőle.

Server
(Kiszolgáló)

A szerviz rövid neve: LanmanServer
Az alkalmazás neve: srvsvc.dll (svchost.exe)
Függés: -
Függesztés: Computer Browser, Distributed File System Remote Installation
Porthasználat: TCP: 139, 445; UDP: 137, 138, 139, 445
Alapértelmezett indítás: automatikus

Ez a szerviz szintén fontos, az adott gép erőforrásainak megosztásához szükséges, konkrétan RPC támogatást nyújt az állomány-, nyomtató- és named pipe (alkalmazások közötti) megosztásokhoz a hálózaton keresztül.
Ha letiltjuk, ezekhez a megosztásokhoz nem lehet hozzáférni, ám nem léteznek megosztások, vagy nincs is hálózatban a gép, akkor nincs rá szükség.

Shell Hardware Detection
(Rendszerhéj hardverfigyelése)

A szerviz rövid neve: ShellHWDetection
Az alkalmazás neve: shsvcs.dll (svchost.exe)
Függés: Remote Procedure Call
Függesztés: Windows Image Acquisition (WIA)
Porthasználat: -
Alapértelmezett indítás: automatikus

Szép nevű, ám funkciójában inkább hétköznapi szolgáltatásról van szó. Az automatikus indítás (AutoPlay) opcióhoz nyújt támogatást, amely képek, zene vagy videóanyagok érzékelését és automatikus megnyitását jelenti egy eltávolítható médiáról/eszközről. Azaz ha pl. egy MP3 lejátszót vagy egy digitális fényképezőgépet bedugunk az USB portba, akkor annak tartalmát rögvest megpróbálja lejátszani a hozzárendelt alkalmazás segítségével az operációs rendszer ezen szervizen keresztül. A szerviz rengeteg médiát/eszközt támogat: pl. Zip és Jaz meghajtókat, CompactFlash, SmartMedia és Memory Stick kártyákat vagy éppen egyéb PC kártyákat, USB és IEEE1394 eszközöket. Tartalom szerint pedig: képeket, (.jpg, .bmp, .gif, .tif), audio anyagokat (.mp3, .wma) és videóanyagokat is (.mpg, .asf).
Ha leállítjuk vagy letiltjuk, akkor elveszítjük ezt a „hardveres" automata lejátszás funkciót, viszont más ismert hatása nincs.

Gál Tamás
MCT, MCSE, MCSA, MVP
gtamas@tjszki.hu

A cikkben szereplő URL-ek:

[1] http://tinyurl.com/6p6cy
[2] http://tinyurl.com/5vx5y
[3] http://tinyurl.com/88rnu
[4] http://support.microsoft.com/?kbid=192806&sd=RMVP
[5] http://blogs.msdn.com/aaron_margosis/default.aspx

IT folyamatok a hazai gyakorlatban

ÉRTÉKELÉS

Az elmúlt években sok (és sokféle) vállalatnál megfordultam, és azt tapasztaltam, hogy nemcsak az egyes szervezetek adottságai térnek el egymástól, hanem azt is, ahogyan ezeket kihasználják, ahogyan élni tudnak a lehetőségekkel. Idővel rá kellett jönnöm, hogy tökéletes vállalat nincs, mindig, mindenhol van mit javítani.

Mottó: „Nem szükséges változtatni. A túlélés nem kötelező."
(W. Edwards Deming, 1900-1993)

Racionális gondolkodásom nem fogadta el elegendő bizonyítékként a saját tapasztalataimat, mások segítségére is szükségem volt ahhoz, hogy általános képet kaphassak a mai, valódi hazai helyzetről. Ez motivált egy kérdőív elkészítésére, amelyben különböző kérdéseken keresztül arra próbáltam választ találni: mennyire szervezettek az IT folyamatok és csapatok, mekkora hangsúlyt fektetnek a cégek a minőség javítására, stb.
A kérdőívre összességében több mint 50 válasz érkezett, különböző méretű és típusú vállalatok különböző beosztású dolgozóitól. **[1]**

Szervezeti felépítés

A felmérésben részt vevők munkahelyének szervezeti felépítésére vonatkozó információkhoz három kérdéssel igyekeztem közelebb kerülni: a vállalat mérete, valamint a válaszadó beosztottainak és feletteseinek száma alapján csoportosítottam.A szervezet méretét tekintve csaknem egyenletes eloszlást kaptam a 10 főnél kisebb létszámtól egészen a 200 főnél többet foglalkoztató mamutvállalatokig. A legkevesebb válaszadó (15.38%) a 25-50 fős intervallumból, míg a legtöbb (28.85%) az 50-200 fő közötti sávból volt.
A felettesek és beosztottak száma közötti összefüggést megvizsgálva arra a következtetésre jutottam, hogy a válaszok nagy része a középvezetők és beosztottak köréből került ki – ennek a ténynek a későbbiekben lesz jelentősége, amikor azt vizsgálom, mennyire van rálátásuk az egyes folyamatokra.

Minőségügyi tanúsítványok

Mint azt már többször hangsúlyoztam, a tanúsítványok megléte nem ekvivalens a minőséggel – ugyanakkor a piaci pozíció megőrzéséhez ma már egyre fontosabb, hogy rendelkezzünk valamilyen „papírral".

Az ide vonatkozó kérdéssel az ISO 9000, illetve a CMM és CMMI tanúsítványok arányát kívántam felmérni. Az eredményt az alábbi diagram szemlélteti:

Minőségügyi tanúsítványok aránya

A kimutatásból jól látható, hogy a szervezetek csaknem fele nem rendelkezik minőségügyi tanúsítvánnyal – akik pedig igen, azok jelentős része ISO 9000 szerint auditált szervezet.

Szoftverfejlesztési módszertanokra vonatkozó ismeretek

A kockázatkezelés témakörében megfogalmazott állítások némileg megosztották a válaszadókat, ám összességében elmondható, hogy a legmagasabb arányú szavazatokat a kockázatkezelésre valóban igaz állítások kapták:

- A kockázat olyan jövőbeli esemény, amely a projekt sikerességét befolyásolhatja.
- Az azonosított kockázatok bekövetkezése nem szükségszerű.
- Ha nem készülünk fel valamely kockázatra, kellemetlen meglepetések érhetnek.

- A kockázatkezelésben felhasználhatjuk korábbi tapasztalatainkat.
- A kockázatok felmérése és kiértékelése a projekt során folyamatos.
- Egy kockázati állítás „HA ... AKKOR ..." formájú kijelentésekből állhat.

Meglepő módon azonban néhány nyilvánvalóan hamis állításra is érkezett szavazat – ezt a válaszadók vállalaton belüli pozíciójának tudhatjuk be. Senki nem akadt, aki valamennyi helyes állítást bejelölte volna – a legtöbben (46.15%) 2 állítást ismertek fel, 3 vagy annál több helyes válasz mindössze 15.38%-tól érkezett. **[2]**

Hasonló a helyzet a konkrét módszertanokra vonatkozó ismeretekkel is: az MSF (Microsoft Solutions Framework), a RUP (Rational Unified Process) és az XP (Extreme Programming) kérdéseire nagy százalékban nem válaszoltak, illetve a választ adók közül igen nagy volt a hibázók aránya: **[2]**

Helyes	Helytelen válasz	Nem válasz	válaszolt
MSF	25.00%	32.69%	42.31%
RUP	19.23%	50.00%	30.77%
XP (alaptényezők)	1 találat: 36.46% 2 találat: 9.62% 3 találat: 3.85% 4 találat: 0%	25% (0 találat)	23.08%
XP (40 órás munkahét)	51.92%	25.00%	23.08%

A (belső) képzésekre vonatkozó kérdések

A szervezetek életében a képzésekre fordított idő és energia mindig kritikus kérdés: az oktatás, továbbképzés ugyanis sokak véleménye szerint (felesleges) költség, azt azonban nem veszik figyelembe, hogy ugyanakkor egyfajta befektetés is – befektetés a jövőbe.

A probléma fontossága miatt a kérdőív számos pontja foglalkozik ezzel a témával, hogy minél finomabb képet kaphassunk.

A kialakult képzési rendszerre vonatkozó két kérdés („Kialakult, hatékony belső képzési rendszerünk van", illetve „Kialakult képzési rendszerünk van, de az egyáltalán nem hatékony") értékelésében azt láthatjuk, hogy mindkét esetben az 1..3 intervallumon találhatjuk az értékek maximumát[1], ami sajnos arra utal, hogy túlnyomórészt egyáltalán nincs kialakult belső képzési rend a szervezeteknél.

Ahol azonban már kialakításra került ilyen rendszer, némileg nagyobb tábort képviselnek a hatékonyságra voksolók (mintegy 0.86 ponttal magasabb értékkel).

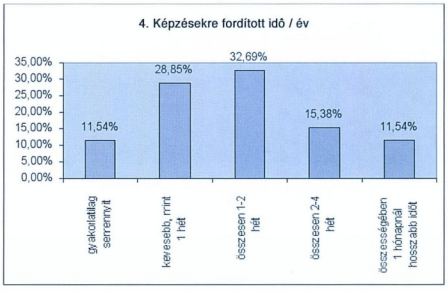
A képzések kialakításával kapcsolatban a jelenlegi, illetve a várható jövőbeli projektek figyelembe vételére kérdeztem rá. Az előbbi egyfajta „tűzoltás"-jellegű megközelítés, amellyel arra próbáljuk megtanítani (továbbképezni) az embereket, amivel éppen foglalkoznak. Nyilván szükség van az ilyen jellegű tanfolyamokra, előadásokra, szemináriumokra is, ám sokkal inkább hosszú távú, stratégiai gondolkodásmódot tükröz a második típusú képzés, vagyis amely során elsősorban a jövőre tervezünk. Az értékelés alapján azt mondhatom el, hogy napjainkban a szervezetek elsősorban a jelenre koncentrálnak. Ennek oka – feltételezésem szerint – kettős lehet: egyrészt sokkal egyszerűbb a „mának élni", mint hosszú távú terveket készíteni; másrészt a rohanó világ, a megrendelők hozzáállása is sokkal inkább ezt a megközelítést preferálja (az igények igen gyakran változnak).

Az alábbi táblázat a hosszú- illetve rövid távra tervezők arányát mutatja, feltételezve, hogy „nagy hangsúlyt" a 6-nál nagyobb értékek jelentenek a kérdőív során használt 10-es skálán:

Hangsúlyos:	
Rövid távú képzési stratégia:	32.62%
Hosszú távú képzési stratégia:	25.00%
Együtt mindkettő:	17.31%

A képzések szervezése mellett fontos még az is, hogy a résztvevőknek milyen visszacsatolási lehetőséget biztosítunk, illetve ők mennyire élnek ezzel. Sajnos ez a két tényező mindössze középszerűnek mondható a felmérésben részt vevők körében (van visszacsatolási lehetőség: 5.65; élnek vele a résztvevők: 5.35), vagyis itt még van mit fejleszteni.

Ebbe a kérdéscsoportba tartozik az éves szinten képzéseken, tanfolyamokon, konferenciákon eltöltött időre vonatkozó pont, amely szintén érdekes eredményt hozott:



Képzésekre fordított idő/év

Az emberek több mint 70%-a tehát idejének kevesebb mint 4%-át tölti továbbképzésen – pontosabban *szervezett* továbbképzésen. Sajnos, az önképzés arányairól nem rendelkezem adatokkal.

Az emberek nyitottsága, vállalkozókedve

Egy informatikai cég esetében az új dolgokra való nyitottságot alapvetően két nagy kategória köré lehet csoportosítani: az új technológiák, illetve új feladatkörök iránti érdeklődésre. A kérdőívben az ennek megfelelő kérdésekre 5.40, illetve 5.94 pont adódik – érdekes tény, hogy az új technológiákat jobban szeretik az emberek, mint az új feladatokat. Az egyes válaszadók körében a két kérdésre adott pontszámok közötti átlagos eltérés 1.37 – a maximum 7 (új feladatkör: 9 pont, új technológia: 2 pont), és a válaszok csaknem egyharmadában a két érték megegyezik. A mérleg tehát többnyire nem, vagy csak kis mértékben billen egyik vagy másik irányba, ami

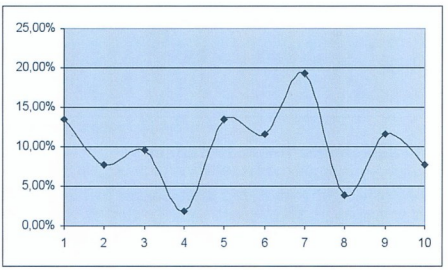
[1] Az értékelő jellegű kérdésekre minden esetben 1 és 10 közötti pontszámok adhatók, ahol 10 jelenti az abszolút egyetértést, 1 a teljes egyet nem értést.

arra utal következtetni, hogy a két tényező szorosan összefügg: az új feladatkör hozhat új technológiai igényt, és a technológiaváltás is járhat új szerepkör betöltésével.

Kommunikáció

Közhely, mégis igaz állítás, miszerint a jó kommunikáció fél siker. Ez egyaránt vonatkozik az ügyféllel történő kapcsolattartásra, ám a vállalaton belüli, kollégák közötti párbeszédekre legalább annyira. Ezeknek a megnyilvánulása természetesen rendkívül sokféle lehet: a hivatalos levelezéstől kezdve a megbeszéléseken, telefonkonferenciákon át a konyhai-folyosói beszélgetésekig széles a skála. Mindezek szerepe azonban ugyanaz kell legyen, a konkrét formától függetlenül: a szervezet működésének, összetartásának, hatékonyságának javítása. A kérdőívben ezzel kapcsolatban éltem az úgynevezett „negatív megfogalmazás" eszközével: a szakemberek szerint ugyan kerülni kell ezt a formai megnyilvánulást, hiszen már ezzel is bizonyos előítéleteket sugárzunk a válaszadók felé, ám célom pont az volt, hogy felhívjam a figyelmet arra: a kommunikációs nehézségek is számos probléma forrásaként szolgálhatnak!
A válaszok tulajdonképpen három nagy csoportra oszthatók, melyek között éles határ húzódik:

- 4 pont alatt helyezkedik el a vállalatok csaknem 30%-a, vagyis rájuk igaz az, hogy nem nehézkes a kommunikáció, jól mennek a dolgok, hatékonyan tudnak együttműködni.
- 4 és 8 pont között a megkérdezettek 50%-a található, itt tehát vegyes tapasztalatokról, megoszló élményekről és véleményekről beszélhetünk (megjegyzésként többször előfordult: „attól függ...")
- 8 pont felett mintegy 20% található, itt egyértelműen nehézkes, akadozó a kommunikáció, az embereknek nehézségeik vannak az együttműködéssel, gyakran még az egymás mellett ülők sem tudják, mivel foglalkozik a másik.

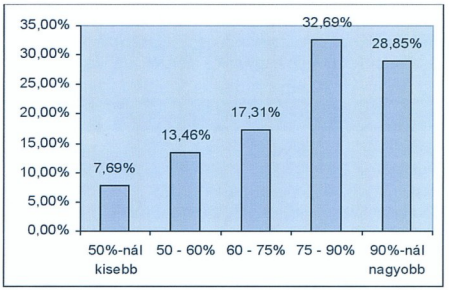


Nehézkes a vállalaton belüli kommunikáció

A fejlesztők kihasználtsága

Az emberi erőforrások kihasználtságának szempontjából kétféle megközelítés között kell megtalálnunk az egyensúlyt. Egyrészt, ha az üzletmenet közvetlen, rövid távú érdekeit nézzük, nyilván a minél jobb kihasználtság az előny, vagyis ha az a 100% felé konvergál. A másik oldalról azonban a folyamatos, magas szintű leterheltség hosszú távon nincs jó hatással a minőségi munkavégzésre, az emberek egy idő után fáradnak, kimerülnek. Ezen kívül az ön- és továbbképzésre is biztosítani kell időt és lehetőséget számukra, hiszen – mint azt már említettem – ez egyfajta befektetés, ami a jövőben többszörösen megtérül.

Végül egy gyakorlati szempont is a leterheltség csökken(t)ése mellett szól: egy projekt alapú szervezetben igen nehéz az erőforrásokat úgy ütemezni, hogy mindenkire folyamatos, egyenletes terhelés jusson, és a folyamatokba is célszerű tartalékokat beépíteni.
Mindezen szempontokat figyelembe véve úgy gondolom, az optimális terhelési arány 60-70% körül mozog: így a szervezet hatékonyságán sem esik csorba, és a megfelelő szabadidőt is biztosítani tudjuk (amely képzésekre, stb. fordítható).
Sajnos a felmérés ennél rosszabb eredményt, magasabb terheltséget mutat: a fejlesztők több mint 60%-a 75%-nál jobban terhelt, sőt a 90% feletti terheltséggel élők majdnem 30%-ot képviselnek!



A fejlesztők átlagos kihasználtsága

Projektirányítással, minőségbiztosítással kapcsolatos fogalmak ismerete

A kérdőívnek ez volt az egyetlen úgynevezett „kifejtős" kérdéscsoportja: a válaszokat nem előre adott halmazból kellett meghatározni, vagy pontszámokkal értékelni, hanem önálló megfogalmazásokat vártam az alábbi négy fogalomra:

- projekt
- projektmenedzsment
- minőség
- minőségbiztosítás

Azzal, hogy nem adtam támpontot a válaszokhoz, célom az volt, hogy a kérdőívet kitöltők teljes mértékben saját gondolataikat fogalmazzák meg.
A *„projekt"* fogalmát viszonylag jó arányban sikerült meghatározni:

Projekt-tényező	Arány
meghatározott cél (termék vagy szolgáltatás)	51.82%
határidő	21.15%
költségkeret	17.30%
ideiglenes szervezet	9.62%

(válaszadók aránya: 61.54%)

A felsorolt 4 projekt-tényező mellett többen utaltak még a minőségre, mint alapfogalomra, illetve a megrendelő szükségességére is. Sajnos többeknek igen rossz véleménye van a projektekről és a tapasztalt projekt-kultúráról („valami, ami a vezetőknek azonnal kell, a szakemberek véleménye meg nem érdekes").

A „*projektmenedzsment*" önmagában egy kétértelmű fogalom: jelentheti azt az irányító-koordináló szervezetet, akit magyarul „projektvezetőségnek" hívhatunk, ám az irányításhoz, összefogáshoz kapcsolódó tevékenységeket is. Ez a kettősség a válaszokban is tükröződött, sőt sokan egyik kategóriába sem sorolták, hanem egy eszköztárnak, módszertannak tartják a fogalmat.

Projektmenedzsment jellemző	Arány
tevékenység (ütemezés, összefogás, irányítás, koordinálás stb.)	34.62%
személy/szervezet, szerepkör	15.38%
eszköztár	5.77%
célja a siker elérése	23.08%

(válaszadók aránya: 61.54%)

A következő vizsgált fogalom a „*minőség*". A kapott eredményt az alábbi táblázat foglalja össze:

Minőség jellemző	Arány
a termékkel/szolgáltatással kapcsolatos követelményeknek való megfelelőség	25.00%
objektív, az adott projekttől független követelményeknek való megfelelőség	21.15%
ügyfél-elégedettség	13.46%

(válaszadók aránya: 61.54%)

A „*minőségbiztosítás*" valaki szerint egyetlen szóval megfogalmazható: „pénz". Az emberek többnyire három nagy kategóriába sorolták a fogalmat: keretrendszernek, tevékenységnek vagy egyszerű ellenőrzésnek tartják. A termék- és a folyamatminőség különbségére is felfigyeltek néhányan, sőt többen megfogalmazták azt is, hogy a minőségirányításnak nem szabad elválnia a cég irányításától.

Minőségbiztosítás jellemző	Arány
keretrendszer, módszertan	15.38%
tevékenység	23.07%
ellenőrzés	15.38%
termékminőség	13.46%
folyamatminőség	7.69%

(válaszadók aránya: 57.69%)

Összességében tehát az fogalmazható meg, hogy az emberek nagy része (60% körül) megpróbálta megfogalmazni ezeket a fogalmakat, melyek gyakran még a szakemberek számára is problémát okoznak: például nem véletlen a sokféle projektdefiníció.

Nemrégiben egy projektmenedzseri konferencián több éve-évtizede a szakmában dolgozó szakemberek (akik mögött többszázmilliós-milliárdos projektek állnak) ebéd közben arról vitatkoztak, hogy tulajdonképpen mi is a projektmenedzsment. Habár számos jellemzőjét sikerült összegyűjteni, megállapodásra nem sikerült jutniuk.

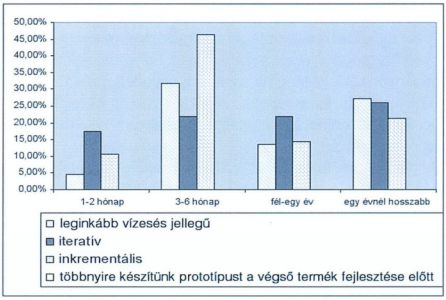
Fejlesztési projektek általános jellemzői

A kérdőívben felmért projektek alig több, mint 10 százaléka mondható igen kis, 3 hónapnál rövidebb átfutásúnak. Az e fölött megadott intervallumok eloszlása 20% és 35% között oszlik meg.

Sajnos a projektek átlagos szervezettsége (terjedelemtől függetlenül) nem mondható túl jónak. Először is, még napjainkban is sok a vízesés-jellegű folyamat – ami kis terjedelmű projektek esetén elfogadható lehet, ám itt az derült ki, hogy még az egy évnél hosszabb projektek 27%-a is ilyen jellegű!

A másik tényező, ami a kérdőív tanúsága szerint hiányzik a magyar szoftverkultúrából, az a prototípus készítésének gyakorlata (vagy igénye). Ennek oka elsősorban az lehet, hogy a megrendelők nem érzik hasznosnak, csupán a költségnövelő mivoltát látják – a fejlesztő cégek pedig általában olyan határidők szorításában dolgoznak, amelynek betartásához rendkívül szoros, feszített tempóra van szükség, amely nem engedi meg a „kísérletezés" luxusát.

A látszólagos többlet energia-befektetés azonban az esetek jelentős részében gyorsan megtérül: a prototípus készítése során ugyanis olyan problémák kerülhetnek felszínre, amelyek később igen nagy kockázatot jelenthetnek. A korai felismerés viszont nemcsak a feltárás, de a megoldás költségét és időtartamát is jelentősen csökkentheti – a projekt egészére nézve így nyereségről beszélhetünk.

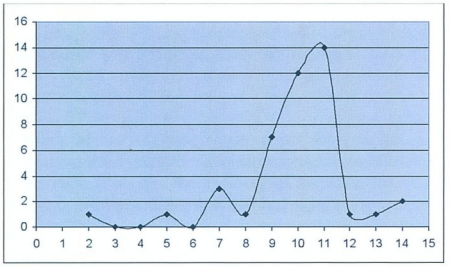


Összefüggés az egyes projektjellemzők között

A folyamatok szervezettsége

Mint azt fentebb említettem, a válaszok jelentős része arra enged következtetni, hogy a hazai vállalatok nem állnak túl jól a szervezettség kérdését illetően. Két kérdés erre közvetlenül is rákérdez, és az itt kapott pontok valamivel fényesebb képet mutatnak: a „jól meghatározott folyamatmodell" 5.13, míg az „ad hoc módon történő fejlesztések" 4.83 pontot kaptak.

Első ránézésre e két értékből arra következtethetnénk, hogy a válaszadók konzekvensen értékeltek: ha X pontot adtak a szervezettségre, akkor 10-X pontot kapott az ad hoc megoldás. Sajnos azonban ez nem így történt: a két érték összegének átlaga ugyan 9.91 (10-hez igen közeli érték), ám a kapott pontszámok 2 és 14 között szóródnak!

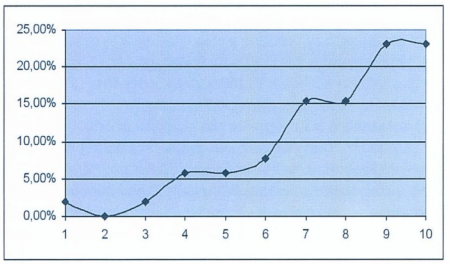


A szervezett és ad hoc folyamatok pontszám-összegének szórása

Mint az a diagramon jól látható, a görbe maximuma valóban 10 körül található, ám a válaszok jelentős része (20%) legalább 2 ponttal eltér a várt eredménytől. Ez sajnos annak tudható be (véleményem szerint), hogy a válaszadók egy része nincs teljesen tisztában a náluk zajló folyamatokkal, azon jellemzőivel – ők azok, akik „gépiesen" végzik munkájukat.
Részben ehhez a kérdéskörhöz kapcsolódóan azt is megállapíthatjuk, hogy a szervezetek nagy része (elvárásainknak megfelelően) projekt-alapú működést mondhat magáénak.

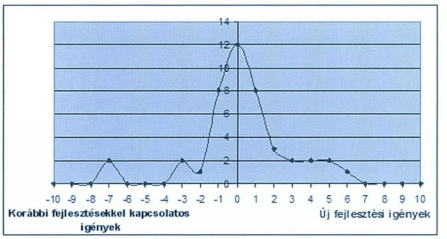
A megrendelői igények

Nemcsak a projekt-alapú, de minden más profitorientált szervezet működésének is alapja az ügyfél, aki igényei kielégítéséért fizetségben részesít bennünket. Folyamataink tehát sohasem öncélúak! Még ha saját, úgynevezett „belső" projektjeink futnak is, azok felfoghatók úgy, mintha mi magunk lennénk a saját ügyfelünk, a „fizetség" pedig a hosszú távon történő megtérülés.
A megrendelőknek tehát igényei vannak: leggyakrabban ezek konkrét, a projekt tárgyát képező elvárások – nem ritka azonban az sem, amikor az ügyfél magába a folyamatba is beleszól. Ilyenkor többnyire ő maga is részese a fejlesztésnek, bevonjuk a megvalósításba, ezért jogos, hogy megfogalmazza ezzel kapcsolatos igényeit is.
A kérdőívben megkérdezettek többnyire a megrendelő igényeihez igazított folyamatokat látnak: erről tanúskodik a 7.75-ös átlagpontszám, és a kérdéshez tartozó válasz-görbe alakja is:



Megrendelőhöz igazított folyamatok

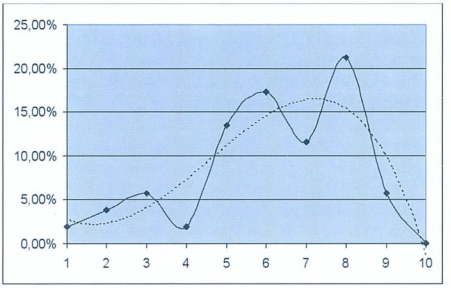
A megrendelő ugyanakkor nemcsak azt határozza meg (legalább részben), hogy hogyan zajlanak a folyamatok, hanem természetesen azt is, milyen végeredményt kell elérnünk. A felmérés válaszai szerint az új fejlesztési igények (7.37 pont) és a korábbi fejlesztésekkel kapcsolatos javítási, bővítési, modernizálási igények (7.06) csaknem azonos arányban jelennek meg a megrendelések között.
Ha azonban az egyes válaszokat külön-külön vizsgáljuk, azt láthatjuk, hogy vállalaton belül általában az új fejlesztési igények vannak túlsúlyban:



A fejlesztési igények megoszlása

Tesztelés

A következő kritikus kérdéskör a tesztelés: ugyan senki nem vitatja a hibajavítás aránylag magas költségvonzatait, mégis kevesen ismerik fel az alapos tesztelés előnyeit. A kapott válaszok némileg magasabb pontszámokat tükröznek ugyan az elvártnál, ám ez szintén adódhat a válaszadók csapaton belüli pozíciójából és elfogultságából egyaránt („én fejlesztő vagyok, jól dolgozom, miért kell ennyi időt elpazarolni a tesztelésre?"). A teszteléssel kapcsolatosan három kérdést fogalmaztam meg, amelyek a fejlesztői, az üzleti és az üzemeltetői tesztek fontosságára vonatkoztak. A részletes értékeléstől eltekintve most csupán az összesítést ismertetem, amely teljes képet ad a felmért tesztelési kultúráról. Az alábbi ábra a válaszok eredményéül kapott görbét mutatja, illetve az ehhez tartozó trendet: a mérleg átbillen ugyan az értékek felső tartományába, ám a csúcs a 7 körüli értékeknél található (6-8), sőt 9 vagy 10 pontot mindössze a válaszadók 5%-a ért el – a kérdőívet kitöltők csaknem 15%-a pedig 5 pont alatti átlagot! Ez pedig azt jelenti, hogy igencsak van még mit javítani a folyamatokon.



Tesztelés

Összegzés

Napjaink IT vállalatai csak úgy tarthatják meg piaci pozíciójukat, ha megfelelő figyelmet fordítanak folyamataik minőségbiztosítására: összefogott, hatékony munkavégzésre van szükség. Ehhez azonban jól kell ismerni adottságainkat, lehetőségeinket és képességeinket, sőt az adott piacot is, ahol fenn szeretnénk maradni – sőt, esetleg erősödni.
Kérdőívem célja ebből eredően kettős volt: egyrészt jómagam szerettem volna információhoz jutni a hazai vállalatok helyzetéről, másrészt reményeim szerint kérdéseim gondolatébresztőként szolgálhattak mások számára is.
Úgy érzem, célom sikerült elérnem: nemcsak összehasonlítási alapot kaptam, hanem értékes véleményeket, hozzászólásokat is. Az ember nem élhet elszigetelten, ismernie kell környezetét ahhoz, hogy értékelni tudja saját és közvetlen környezete helyzetét. A beérkezett válaszok megerősítettek abban, hogy van mit javítani – ám a helyzet nem reménytelen.

Megjegyzés

A kérdőív publikálása levelezőlistákon, illetve személyes weboldalamon **[3]** történt, így pontosan nem tudom megmondani, kikhez jutott el. Néhány esetben a megjegyzés rovatban kaptam utalást arra, hogy néhányan önszorgalomból is terjesztették az elérhetőségét. Többen feltüntették e-mail címüket vagy egyéb elérhetőségüket, hogy az eredmény elkészülte után tájékoztassam őket – természetesen ezeket az adatokat nem vettem figyelembe a kiértékelés során, és a megjelöltől eltérő célra nem használom fel.

Molnár Ágnes
molnar.agnes@cleware.hu

A cikkben szereplő URL-ek:

[1] A pontos kérdések, és a részletes értékelés: http://www.cleware.hu/agi/kerdoiv
[2] Testreszabott szoftverfejlesztési módszertanok: http://www.cleware.hu/agi/bme/H5IV6V_2004osz.pdf
[3] Személyes weboldal: http://www.aghy.uw.hu

Átkapcsolás a 64-bites megoldásokra

Széles körben elérhetővé váltak a Microsoft Windows Server 2003 x64 változatai és a Windows XP Professional x64 verziója. Az új verziók egyetlen platformon belül lehetővé teszik az új 64-bites alkalmazások, valamint a létező 32-bites alkalmazások csúcsteljesítményen történő futtatását.

A Windows Server 2003 x64 változatainak és a Windows XP Professional x64 változatának kifejlesztése során a Microsoft szorosan együttműködött az iparág vezető képviselőivel, köztük több chip- és hardvergyártóval. Ennek eredményeként az új operációs rendszer a 64-bites Intel Xeon és Intel Pentium 4 processzorokon ugyanúgy fut, mint az AMD Opteron és Athlon 64 processzorokon. Az Intel Itanium processzor alapú rendszereken futó Windows Server 2003 továbbra is a legrobosztusabb Windows platform.
Az AMD Opteron és Intel 64-bit Xeon processzorokat támogató SQL Server 2000 Service Pack 4 is elérhető. Az új funkcióknak köszönhetően a felhasználók 32-bites alkalmazásokat futtathatnak a 64-bites architektúrán. A Service Pack 4 kihasználja az adatbázis és a processzor architektúra között optimalizált erőforrás-gazdálkodás és teljesítmény szinergiát, annak érdekében, hogy az SQL Server 2000 felhasználói számára fokozott méretezhetőséget biztosítson. Az év második felére tervezett SQL Server 2005 szintén támogatni fogja az AMD Opteron és Intel 64-bites Xeon processzorokat, hogy a felhasználók egyszerre futtathassanak nagy teljesítményű 32- és 64-bites megoldásokat.

A Windows Server 2003 x64 változatai és a Windows XP Professional x64 változata 32-bites megfelelőikkel azonos áron érhetők el. Azon ügyfelek, akik a Windows megfelelő 32-bites változatát x64 hardverrel együtt szerezték be, jogosultak azt az új x64 változatra cserélni. A Microsoft mennyiségi licencelési ügyfelei ezt az adathordozó-készlet segítségével tehetik meg. Azon ügyfelek, akik a Windowst egy OEM-től vagy rendszerépítőtől szerezték be, az új szoftverhez a Technology Advancement Program keretében juthatnak hozzá.

További információ:

http://www.microsoft.com/windowsserver2003/howtobuy/licensing/pricing.mspx
http://www.microsoft.com/x64

VPN karantén

Windows Server 2003 + ISA Server 2004 környezetben

A VPN csatlakozások számának növekedésével a VPN karantén alkalmazása egyre többször kerül szóba.

A karantén kissé rosszemlékű szó. Nekem mindig Rejtő Jenő egyik nagyszerű könyve ugrik be erről a kifejezésről: *„A karanténban lehorgonyzott járművekre egy kis őrhajó felügyel, amely háromóránként kifut a tengerre, és sorra veszi a járműveket. Az örök errefelé kérdezés nélkül lőnek, akár a partról, akár a tenger felől közeledik valaki."* (Rejtő Jenő: Az elveszett cirkáló).

Ám egy ideje már valami más, lényegesen földhözragadtabb téma is beugorhat minden üzemeltetőnek e kifejezéssel kapcsolatban. Pontosabban főleg azoknak, akik hozzáférést adnak azoknak az otthoni/utazó felhasználóiknak, akik nyilvános hálózaton keresztül férnek hozzá a cég belső hálózatához. Ha nem csak levelezésről, OWA-ról van szó, akkor célszerű VPN segítségével csatlakozniuk, hiszen maga a csatlakozás és a „csőben zajló" kommunikáció is biztonságos, ráadásul viszonylag egyszerűen megvalósítható mindkét oldalon illetve könnyen is szabályozható (többféle protokoll, házirendek, névfeloldás, stb.). De van egy súlyos hátránya is, mert bár a VPN ügyfél a sebességet leszámítva úgy érzi ugyan, mintha benti hálózatban dolgozna, és ennek szerfölött örül(het) is, ám mi igen rosszul érezhetjük magunkat, és szomorúak vagyunk: tudnillik nem hat rá semmilyen biztonsági óvintézkedés, ami bent megszokott. Nincs pl. központilag kötelezően bekapcsolt tűzfala, nincsenek csoportházirendből érvényesülő megszorítások, nincs központi helyről automatikusan frissülő, víruspajzzsal ellátott víruskereső, nincs a rendszergazda által feltelepített, háttérben settenkedő antispyware program. Ezen kívül fogalmunk sincs, honnan kapcsolódik, és mi minden van feltelepítve a gépére, és még sorolhatnánk. De bent van a hálózatban, terjesztheti a kártevőket, lehet rajta akár több vírus/spyware szóró automata, meg spéci SMTP szerver, futtathat bármit (akár a tudta nélkül is), hiszen egy átlagos felhasználótól nem várható el, hogy odafigyeljen ezekre a dolgokra. Ez így viszont nem túl ésszerű, hiszen, hogy védjem meg a hálózatomat, ha egyrészt adni akarok (muszáj) lehetőséget a távoli munkára, hozzáférésre, de a másik kezemmel meg olyat engedek meg, amit a belső hálózaton a legnagyobb jókedvemben, a legtöbb kólát/sört/hamburgert/pizzát hozó „józernek" sem? Pedig így van, ez a dolog benne van a pakliban.

De azért van némi gyógyír,

ez pedig a VPN karantén, amelyet már az ISA 2004 előtt is kipróbálhattunk, hiszen a Windows Server 2003 a Resource Kit Tools-sal és az RRAS-sal felturbózva kínál egy megoldást. Ám az ISA 2004-gyel mégis komplexebb (nemcsak port- és időtartamszűrő) és mégis egyszerűbb módszerhez jutottunk, ezért főképp ezt a módszert vizsgáljuk meg. Természetesen nem tökéletes és hibátlan ez a technológia sem, de erről majd később lesz bővebben szó.

A cikk kifejezetten gyakorlati szempontból közelít a témához, méghozzá 4 fő részre szelve azt:

- Előkészületek és működés
- Karantén beállítása az ISA szerveren
- Profil létrehozása a Connection Manager Administration Kit-tel (Csatlakozáskezelő felügyeleti csomag)
- Profil kijuttatása az IIS/ISA párossal, illetve némi kliens oldali teendő

Tehát először essen pár szó a karantén létrehozásának feltételeiről és működésének alapjairól.

Előkészületek

- Nincs szükség különleges előkészületekre, és elvileg még a Windows Server 2003 sem szükséges, mivel az ISA 2004 lényegében „hozza" a szolgáltatást (igaz, a Windows 2003 Resource Kit Tools telepíthető, de a szükséges szerviz telepítése sikerült, tovább viszont nem mentem ebben a környezetben). A Windows Server 2003 SP1 az ISA 2004-es megvalósítás esetén sem életbevágó feltétel, önmagában való alkalmazása esetén viszont kifejezetten hasznos, hiszen az SP1 telepítése után már külön szkript nélkül (lásd később) is megoldható a VPN kapcsolat megtagadása, ha a kliensen nincsenek telepítve a legújabb biztonsági frissítések.
- Fontos elem viszont a Windows 2003 Resource Kit Tools **[1]**, pontosabban a következő 4 komponense:
- Rqs.exe: Remote Quarantine Server
- Rqc.exe: Remote Quarantine Client
- Rqs_setup.bat: a Remote Access Quarantine Agent szerviz telepítője/beállítója (a szerveren)
- RqsMsg.dll: Remote Access Quarantine Agent Message DLL

Ezzel kapcsolatban felhívnám a figyelmet arra, hogy a Resource Kit Tools kiadása óta frissítették az rqs.exe-t, amelyet a Resource Kit telepítése után (de csak ekkor) tudunk feltelepíteni, és célszerű is, mert a Microsoft ajánlása szerint az új ISA-hoz ez kell **[2]**.

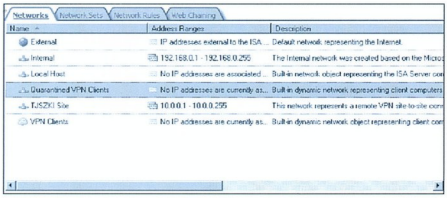
- A kliensekről még nem beszéltünk, viszont bőven nem is kell, Windows 2000-től felfelé a kliens és szerver Windows-ok egyaránt alkalmasak erre a feladatra.

- Természetesen egy korrekt, működő VPN szerverre is szükség van, ISA 2004 alatt ez egyszerűen megoldható, ám erről itt most nem lesz szó, Tom Shinder viszont részletesen kifejti **[3]**.

Hogyan működik?

Eddig kiderült, hogy a Resource Kit ad két fontos komponenst, azaz a karantén ellenőrzési módszerének lényegi elemeit. E módszer szerint, amikor a speciálisan felturbózott VPN kliens csatlakozik, akkor az ISA beengedi, de egyelőre elkülöníti a Quarantined VPN Clients „hálózatba". Ekkor kezdődik az érvényesítés, azaz annak/azoknak a feltételnek/feltételeknek a lekérdezése, amelyet megkövetel(het)ünk a VPN klienssel szemben. Néhány példa erre, jellemző „kérdések" zárójelben:

- tűzfal (be van-e kapcsolva?)
- ICS (ki van-e kapcsolva?)
- javítócsomagok (fent vannak-e a legfrissebbek?)
- víruskereső (van-e? az adatbázis friss-e?)
- jelszó (elég komplex-e?)
- jelszavas képernyővédő (van-e?)
- stb., igény és persze programozói tudásszint szerint



Külön hálózata van a VPN-Q klienseknek

A gyakorlatban ezeknek a feltételeknek a lekérdezője/figyelője az Rqs.exe a szerveren, a megvizsgáló/küldő pedig az Rqc.exe a kliensen. Ha a feltételeknek megfelel a kliens, akkor az utóbbi egy speciális (ún. SIGNATURE) sztringet küld a szervernek és ezután az ISA 2004 kiemeli a karanténból és behelyezi a szimpla VPN kliensek közé.

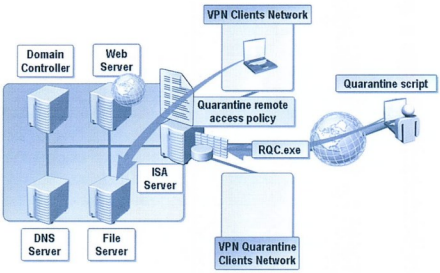
De hogyan kerül a kliensre pl. az Rqc.exe, hogyan és hol fut le az ellenőrzés? Mi alapján ellenőriz? Ezekre a kérdésekre a válasz a CMAK, amely egy jó sok (kb. 20) lépésből álló szerveroldali varázsló. A varázslás során elkészítünk majd egy ún. CM (Connection Manager) profilt, amely egy .exe csomag, és amely magába foglalja a VPN-Q klienst, az általunk legyártott ún. „post-connect" ellenőrző szkripteket (.vbs, .cmd, .exe, .bat), az esetleges nyelvi válaszfájlokat és minden mást, amit le kívánunk küldeni a kliens gépre. Valamilyen úton-módon aztán lekerül (pl. letölti a delikvens a dedikált weboldalunkról), majd elindítja. A csomag a telepítés után rögvest bekéri a felhasználónevet/jelszót, csatlakozni próbál és siker esetén máris bekerül a karanténba. Így kerek a művelet.

Karantén beállítása az ISA szerveren

Tehát fussunk neki végre a gyakorlati tennivalóknak, magán a VPN szerveren, ami a mi esetünkben értelemszerűen az ISA Server 2004.

- Miután feltelepítettük (és az RQS frissítést is), navigáljunk el parancssorban a Resource Kit Tools mappájába!

```
C:\Program Files\Windows Resorce Kits\Tools
```

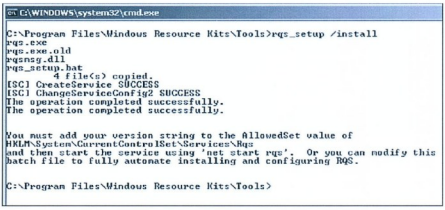


A komplett folyamat

Használjuk a következő parancsot a „Remote Access Quarantine Agent" szerviz telepítésére és beállítására.

```
Rqs_setup /install
```

Ezután a Computer Management MMC-ben a frissen kisütött szolgáltatás induló értékét automatikusról kézire állítsuk át (később lesz értelme). Ha esetleg később meg akarjuk szüntetni a karantén funkciót, akkor a „/remove" paraméter eltávolítja a szervizt és a leendő regisztrációs adatbázis bejegyzéseket egyaránt.



Ahogyan a szkript figyelmeztet is, nyissuk meg a regisztrációs adatbázist és tegyünk két bejegyzést a következő kulcs alá:

```
HKEY_LOCAL_MACHINE \ SYSTEM \ CurrentControlSet \
Services \ rqs
```

Az első általunk elkövetett bejegyzés ezen a kulcson belül egy Multi-String Value típusú legyen, a neve „AllowedSet" tartalma pedig pl. „Version1". Ez az érték egy azonosító, amely a szkriptünk első sorában szintén ugyanezt a tartalmat fogja hordozni. A másik bejegyzés String típus legyen, a neve „Authenticator" az értéke pedig:

```
C:\Program Files\Microsoft ISA Server\vpnplgin.dll
```

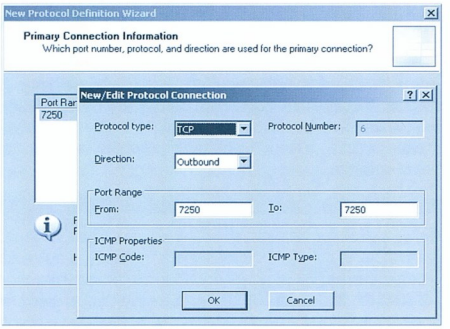
Ez a bejegyzés fogja megsúgni az Rqs.exe-nek, hogy kliensoldali Rqc.exe-vel lefolytatott sikeres párbeszéd után, kinek küldje el a karanténból való kiengedésre szóló felszólítást. Ha például a Windows Server 2003-at használnánk a karantén megvalósítására, az érték a következő lenne.

```
C:\Windows\System32\mprapi.dll
```

A Windows szolgáltatásokról szóló cikksorozatunk figyelmes olvasói nyomban megmondják, hogy ez maga az RRAS ☺. Ezzel a regisztrációs-adatbázisturkálást befejeztük, haladjunk tovább, indítsuk el az ISA Server MMC-t.

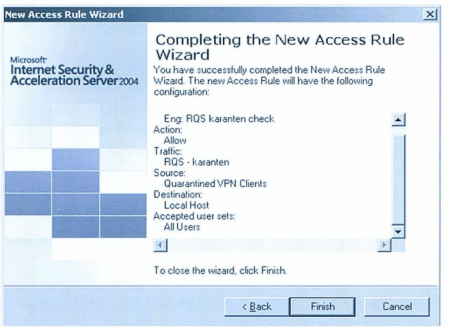
Szükséges készítenünk egy tűzfalszabályt a VPN karanténban leledző klienseknek, azért, hogy a megkövetelt restrikciók teljesítése után a kliens tudja értesíteni a szervert, hogy immár szabad „leszedni" róla a karantént. Mivel alapesetben a VPN karantén ISA „hálózatra" semmilyen tűzfal szabály nem érvényesül, ezért e nélkül a megengedő szabály nélkül ez nem lehetséges. A szabály létrehozásának első lépése, hogy egy az új speciális protokoll készítünk.

- A Firewall Policy-ra kattintva, válasszuk a legszélső panelben (Task Pane) a Toolbox/Protocols elemet, majd New/Protocol.
- A névadás után New, majd adjuk meg a paramétereket (TCP/Outbond/7250-7250) és nem kívánunk másodlagos kapcsolatot létrehozni.



Az új protokoll paraméterei

Ezután jöhet a szabály, azaz lépjünk a baloldali faszerkezetben a Firewall Policy > New Access Rule menüpontra.

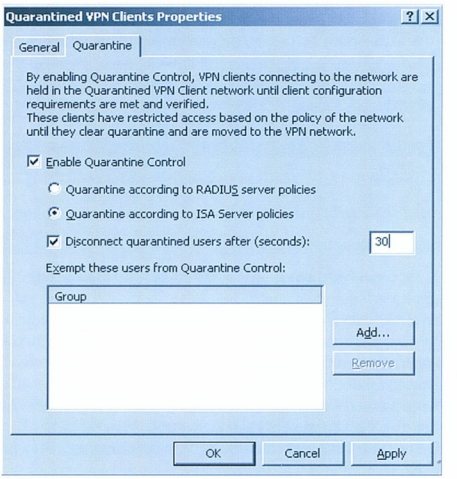


A szabály összegző képernyőjéről minden kiderül

- Megengedő szabály lesz, természetesen az előbb elkészített protokollt használjuk, a forrás a VPN karantén kliensei, a cél pedig az ISA 2004 Server, azaz a Local Host.

Most pedig elárulnék egy „randa" dolgot: az eddigi műveleteket (szerviz telepítés, regisztrációs adatbázis machináció, protokoll definíció, és tűzfal szabály) egyetlen szkripttel is elvégezhetjük, mégpedig a ConfigureRQSForISA.vbs nevűvel, amelyet az útmutató dokumentummal együtt letölthetünk **[4]**. Immár az ISA szerveren sem maradt más dolgunk, minthogy engedélyezzük és beállítsuk a karantént. Ehhez navigáljunk a Configuration/Networks pontba és kattintsunk egy duplát a Quarantined VPN Clients sorra.

Az engedélyezés mellett lehetőségünk van megszabni, hogy mennyi ideig maradhat a kliens a karanténban, mielőtt leválasztja az ISA, valamint megadhatunk olyan felhasználókat, akikkel kivételezünk, azaz ők egyből a „VPN clients" hálózatba kerülnek, mindenfajta vizsgálat nélkül.



A karantén alapbeállításai

Mostanra tehát a karantén kész, az örök felálltak, ám egyelőre még arra is lőnek, amire később már nem kell.

Folytatjuk...

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

A cikkben szereplő URL-ek:

[1] http://tinyurl.com/6p6cy
[2] http://tinyurl.com/dc2u7
[3] http://www.isaserver.org/articles/2004vpnserver.html
[4] http://tinyurl.com/6db8h

Ami a hivatalos Microsoft tanfolyamokból kimaradt...

Active Directory – webes regisztráció (1. rész)

Rovatunk e cikkében olyan weblapot készítünk, melynek segítségével a felhasználók maguk kérvényezhetik hozzáférési szándékukat a rendszerünkhöz, intranetünkhöz. Tehát az önregisztráció alapján automatikusan létrehozzuk a felhasználó account-ját, és szükséges beállításait az AD-ben. Továbbá felépítjük az ellenőrzési, jóváhagyási struktúrát úgy, hogy mindez egy webes felületen könnyedén legyen követhető minden szereplő számára.

Nem egy helyen találkoztam azzal a problémával, hogy olyan regisztrációs eljárásra lett volna szükség, ahol a leendő felhasználók maguk igényelhetik a hozzáférést az intranethez valamilyen egyértelmű, de a lehető legegyszerűbb ellenőrzési forma mellett. Alaphelyzetben egy felhasználó létrehozása az Active Directoryban nem nagy gond, hisz a rendszergazda pár kattintással létre tudja hozni a felhasználó accountját. De mi van akkor, ha a célközönség száma olyan méreteket ölt, amit már kattintgatással nehéz követni, vagy a nagyszámú felhalmozódó igények miatt túl sok az átfutási idő? Ilyen esetben jól jön az automatizmus.

Virtuális város

Ahhoz, hogy érthető legyen a feladat, építsünk fel egy átlátható környezetet elméletben. Képzeljünk el egy átlagos várost (vagy önkormányzatot), ahol informatikai intranet (e-mail, adatbázis, portál, stb. elérési lehetőség) épül az alkalmazottak (tanárok, orvosok, szociális és kulturális intézmények alkalmazottai, és még sokan mások) számára. Mondjuk a város szívében van egy informatikai központ, ahol a Microsoft technológia lakik. Itt gyülekeznek a szerverek, amelyek a már említett e-mail, adatbázis, portál, stb. elérési szolgáltatásokat nyújtják a leendő felhasználók számára. Képzeljük hozzá, hogy a központra online (mondjuk kábeltévé technológián, de legalább ISDN csatlakozással, és persze VPN alapon) kapcsolódik minden olyan intézmény, ami a városkánk fenntartása alá tartozik (tehát iskolák, orvosi rendelők, szociális és kulturális intézmények). Nagyságrendileg legyen ez mondjuk 100 intézmény (épület) szerteszét a városban, és tartozzon ide összességében közel 3000 alkalmazott. Minden egyes épületben álljon rendelkezésre mondjuk 5 munkaállomás a leendő felhasználók számára. Ugye ez azt feltételezi, hogy a felhasználók felváltva fogják használni a gépeket, és mindig azon munkaállomás elé fognak leülni, amelyik éppen szabad az intézményükben. Legvégül, de nem utolsó sorban az egész projekthez rendeljünk hozzá egyetlen egy informatikust a központban, aki az egyéb munkái mellett még e folyamatnak a felügyeletét is megkapja. Röviden: 100 végpont; 500 munkaállomás; 3000 felhasználó; 1 elcsigázott rendszergazda. ☺

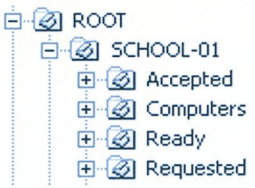
A feladat

Mivel a kábeltévé és az ISDN technológia még viszonylag kis sávszélességet biztosít épületenként, ráadásul még ezen is osztozik az 5 munkaállomásunk, ezért mondanom sem kell, hogy az egységes információs probléma megoldása, továbbá a nagyszámú felhasználó könnyed kezelése bármely végponton, egyértelműen mutatja a kézenfekvő megoldást: webes kommunikációs felület...

Készítsünk tehát egy olyan weblapot az intranetünkre, melyen kitölthetik a leendő felhasználók saját adataikat, igényelt bejelentkező nevüket, elfogadhatják az EULA (etikai kódex esetleg, vagy belépési szabályok) feltételeit. Az igénylésről küldjünk egy automatikus értesítést levélben annak az intézményvezetőnek, akihez az igénylő tartozik. Az intézményvezető saját hatáskörében döntést hozhasson, hogy hozzájárul-e az igénylő regisztrációjához, vagy sem. Erről a hozzájárulásról (vagy „sem"-ről) kapjon egy szintén automatikus levelet az elcsigázott rendszergazda, aki szintén a webes felületen utolsó láncszemként (ha mindent megfelelőnek talál), egy pipálással élesítheti az igénylést. Bár sok mindennel lehetne még finomítani az eljárást (igénylések visszavonása, törlése, felfüggesztése, stb.), de az érthetőség kedvéért maradjunk ennél az egyszerű folyamatnál.

Készítsük elő a terepet

Legelőször is, mielőtt nekilátnánk a regisztrációs weblap elkészítéséhez, készítsük elő az Active Directory-t. Hozzunk létre egy OU-t, amit a fő ágnak tekintünk. Ez alá hozzuk létre az első intézményünk OU-ját. Ezen intézményi OU alá pedig három olyan OU-t, melyekben gyűlnek az igénylések, az elfogadott kérelmek, és az aktív felhasználók. Továbbá szükséges az intézményi OU alá még egy olyan, ami az intézményhez tartozó munkaállomásokat gyűjti. Szükségünk van továbbá az intézményvezető felhasználóra közvetlenül az intézményi OU alá. Valahogy így:



Tehát a „Requested" OU-ba kerülnek az új igénylések, természetesen letiltott állapotban, hisz csak a jóváhagyási procedúra után élesedhet a felhasználói fiók. Az „Accepted" OU-ba a kijelölt vezető által elfogadott és jóváhagyott felhasználók. A „Ready" OU-ba pedig a folyamat végén azok a felhasználók kerülnek, akiket a rendszergazda is kipipál, és ezáltal aktiválódnak a rendszeren. A „Computer" OU-ba pedig a már említett intézményhez tartozó munkaállomások kerülnek.
Ugyancsak készítsünk a jogosultságok könnyed kezelése érdekében biztonsági csoportokat a „ROOT"-ba:

Értelem szerint a „Kérelmezők" csoportba fognak kerülni azok az „account"-ok, akik nevében a webes form-ot kitöltik az igénylést elindító leendő felhasználók. Erre a szerepkörre még a későbbiekben visszatérünk, de most elégedjünk meg ennyivel, hogy a biztonsági csoportot létrehozzuk. A „Vezetők" csoportnak pedig természetesen azon felhasználók lesznek tagjai, akik a jóváhagyási folyamat első szintjén döntést hozhatnak az igénylések elfogadásáról, illetve elutasításáról.

Web.config

Ahhoz, hogy a megoldásunk könnyedén átalakítható legyen saját névtereinkre, vagy akár a későbbi módosítások végett is, az alapinformációkat tegyük ki a web.config fájlba. A web.config fájl az IIS szerverünkön az alkalmazásunknak létrehozott site-ján, annak is a gyökerében helyezkedjen el. Tehát a bejegyzés:

```
<appSettings>
<!-- Az AD-t fenntartó domain gyökerének LDAP
útvonala -->

<add key="ad.domainldap" value="DC=COMPANY,
DC=local"/>
```

Az „add key" tartalmazza a kulcsot, amit a webes alkalmazásunk hivatkozásként fog használni az adatok kiolvasásához, a „value" pedig az értéket. Esetünkben a tartományi nevünk: COMPANY.local.
A már említett automatikus levélküldés alapinformációit is érdemes ide kitenni a későbbi könnyed kezelhetőség érdekében:

```
<!-- Exchange server adatok-->
<add key="exch.server" value="SRV-EXCH-01"/>
<add key="exch.org" value="COMPANY"/>
<add key="exch.storagegr" value="First Storage
Group"/>

<add key="exch.admingr" value="First
Administrative Group"/>

<add key="exch.emaildomain" value="company.hu"/>
```

A kulcsok, és azok értékei magukért beszélnek, ezért nem részletezem külön őket. Folytatván a sort, az intézményi OU struktúrát is kezeljük a konfigurációs fájlban:

```
<!-- A felhasználó állapotát tükröző OU-k nevei
kis és nagybetű helyesen) Ezek az OU-k a rootldap
elérésű ou alatt elhelyezkedő intézményi OU-kban
kell hogy legyenek -->
<add key="ad.requestedou" value="Requested"/>
<add key="ad.acceptedou" value="Accepted"/>
<add key="ad.readyou" value="Ready"/>
```

Ha ránézünk a korábbi ábrára a cikkben, akkor láthatjuk, hogy pontosan milyen elnevezéseket tartalmaznak az értékek. Nem állunk meg, hanem a szükséges biztonsági csoportokat is definiáljuk:

```
<!-- Annak a csoportnak a neve (domain\csoport)
amelyiknek a tagjai új kérelmeket generálhatnak -
>
<add key="auth.requesters"
value="COMPANY\Kerelmezők"/>

<!-- Annak a csoportnak a neve (domain\csoport)
amelyikben az intézményvezetők vannak-->

<add key="auth.institutionchiefs"
value="COMPANY\Vezetők"/>
```

Szükségünk van a főág, más néven a „ROOT" definiálására is, hogy az alkalmazásunk tudja hol helyezkednek el az intézményeink, és azok gyűjtő OU-jai:

```
<!-- Az intézményi OU-k közvetlen szülőjének LDAP
elérési útja a DC-k nélkül -->

<add key="ad.rootldap" value="OU=ROOT"/>

</appSettings>
```

Persze még rengeteg mindent lehet a web.config fájlban definiálni, de ne menjünk nagyon előre. Az induláshoz éppen elég ennyi információ. A lényeg annyi, hogy az értékek egyezzenek az AD-ben definiált csoportok és OU-k elnevezéseivel (kis és nagybetű helyesen). Minden definiált érték az asp.net-ben a System.Configuration.ConfigurationSettings.AppSettings collection-ban megtalálhatóvá válik. Ezt a collection-t nevezzük el egy rövidített formában „cnfg"-nek. Tehát ha a web.config fájlban van megfelelő helyen egy <add key="abcd" value="valami"/> paraméter sor, akkor a cnfg["abcd"] értéke "valami" lesz a hivatkozás felhasználásakor.

A Web form

Természetesen a fenti web.config fájlban meghatározott értékeket mindenki maga változtathatja saját szájíze szerint. Ez rugalmasan kezelhetővé teszi megoldásunkat. Most, hogy a web.config fájllal egyelőre végeztünk, hozzunk létre egy webes form-ot az adatok bekérésére. Hozzunk tehát létre egy form.aspx fájlt, és a szükséges grafikai keretek, és egyéb kötelező csillivilik között kérjük be a leginkább szükséges adatokat, amik az AD „account" létrehozásához szükségesek. Ilyenek a bejelentkező név, teljes név, keresztnév, vezetéknév.
Kezdjük a teljes név bekérésével valahogy így:

```
<TR>

  <TD>

    <asp:label id="txtLabel"
runat="server">Teljes név:

    </asp:label>

  </TD>

  <TD width="100%">

    <asp:textbox id="txtName" tabIndex="1"
runat="server" CssClass="textbox" MaxLength="500"
Width="100%">

    </asp:textbox>

  </TD>

</TR>
```

A „txtLabel" lett a „Teljes név". A „txtName" pedig a kitöltendő értéket fogja tartalmazni. A táblázathoz kapcsolódó stílust szabályzó adatok azt eredményezik, hogy a szövegbeviteli mező kitölti a számára lefoglalt maximális hosszúságot. Ugyanilyen módon hozzunk létre további beviteli mezőket, a bejelentkezési névhez (txtLoginName), a vezetéknévhez (txtLastName) és az utónévhez (txtFirstName). Készítsük el a bevezetőben említett EULA oldalt is, aminek az aljára egy „elfogadom / nem fogadom el" választási lehetőséget tegyünk. Természetesen úgy, hogy a továbblépés csak az elfogadás esetén legyen lehetséges. Itt akár le is zárhatjuk a folyamatot, hisz rendelkezésünkre áll minden szükséges információ.

Az AD account létrehozása

Mivel minden szükséges adatot bekértünk, nincs más hátra, mint létrehozni az AD-ben a bekért adatok alapján az account-ot. Az aspx fájlunkban definiáljunk egy metódust ennek a folyamatnak (mondjuk: ADRequest).

```
private void ADRequest()
{
  UserCreated.ADFacade aUser;
  UserCreated.UserData ud;
  aUser =
(UserCreated.ADFacade)Session["adgateway"];
  ud = new UserCreated.UserData();
  ud.Loginname = txtLoginName.Text;
  ud.FirstName = txtFirstName.Text;
  ud.LastName = txtLastName.Text;
  ud.FullName = txtName.Text;
}
```

A metódusunk mindjárt az elején hivatkozik egy „UserCreated"-re ami egy dll, és a segítségével létrehoz két objektumot.

- Az „ADFacade" dolga lesz elvégezni a az AD műveletet.
- A „UserData" pedig tartalmazza a szükséges adatokat.

Az „ADFacade" típusú „aUser" objektumnak készítünk egy „CreateRequest" metódust, amihez szükséges paraméternek a másik objektum (UserData) szolgál adattal. Ez a belső „CreateRequest" metódus fogja létrehozni az AD „account"-ot a kijelölt helyen, ha minden adat megfelelően rendelkezésére áll. Készítsük el ezt a UserCreated.dll-t, ami egy teljesen normális dll kiindulási helyzetben. Helyezzük el a „site"-unk „bin/" könyvtárába. A már fentebb említett, és a meghívott „ADFacade" metódus által életre keltett objektum belső metódusának (ez a „CreateRequest" metódus) az alábbi fontosabb kódrészleteket kell tartalmaznia feladatához:

```
public string CreateRequest(string instituteDN,
UserData ud)

{

  ...

  DirectoryEntry root, container, user;


  this.lastModifiedUser = ud;


  container = new DirectoryEntry("LDAP://OU=" +
requestedOU + "," + instituteDN);

  user = container.Children.Add("cn=" +
ud.Loginname, "user");

  user.Properties["samAccountName"].
Add(ud.Loginname);

  user.Properties["sn"].Add(ud.FirstName);

  user.Properties["givenName"].Add(ud.LastName);

  user.Properties["displayName"].Add(ud.FullName);

  ...

}
```

Mint fentebb említettem, az „ADFacade" metódus paramétereként a „UserData" objektum szolgál. Tehát le kell kérdeznünk az AD-t a szükséges adatok tekintetében:

```
public UserData getUserData(DirectoryEntry
userObject)

{

...

  UserData retval;

    if (userObject == null)

      {

   return null;

      }
```

```
    else

    {

    retval = new UserData();

    this.lastModifiedUser = retval;


    if
(userObject.Properties["samAccountName"].Count>0)

        retval.Loginname =
userObject.Properties["samAccountName"][0].ToStrin
g();

    if
(userObject.Properties["displayName"].Count>0)

        retval.FullName =
userObject.Properties["displayName"][0].ToString()
;

    if(userObject.Properties["sn"].Count>0)

        retval.FirstName =
userObject.Properties["sn"][0].ToString();

if(userObject.Properties["givenName"].Count>0)

        retval.LastName =
userObject.Properties["givenName"][0].ToString();

    }

...

}
```

A fenti kódrészlet jól mutatja a lekérdezéseket. Ennek alapján, és az előző kódrészletek tanulmányozásával könnyen kiegészíthetjük munkánkat olyan információk felhasználásához, mint mondjuk a lakcím, felettes, stb. információk bekéréséhez, lekérdezéséhez, és a felhasználó létrehozásakor a megfelelő AD helyek kitöltéséhez.
Most fújjuk ki magunkat egy kicsit. Előkészítettük az Active Directory-t a felhasználói igénylések fogadására. Készítettünk biztonsági csoportokat azon vezetők részére, akik jóváhagyási jogosultsági körrel fognak rendelkezni. Átnéztük, hogyan kell készíteni form-ot adatok bekéréséhez webes technológián. Végül, de nem utolsó sorban, készítettünk egy mini dll-t, aminek két metódusa segítségével létre tudunk hozni az AD-ben „account"-ot.
Már csak egy kérdés maradt hátra a regisztrációs folyamathoz, mégpedigen az, hogy miként fér hozzá a kérelmező felhasználó egyszerűen a kitöltendő form-hoz, ha még nincs is hozzáférése a rendszerhez (hisz épp most igényelné). Erre több jó megoldást is el tudok képzelni. Mondjuk kinevezünk egy adott munkaállomást, amin egy felügyelt felhasználó segítségével ki lehet tölteni a form-ot. Vagy esetleg ki lehet rakni egy publikus weboldalra az igénylési lapot, hisz úgyis jóváhagyással lehet csak bekerülni a rendszerre. Mi most maradjunk egy ritkán használt megoldásnál. Tegyük azt, hogy minden intézményi OU alá hozzunk létre (a vezetői felhasználó mellé) egy olyan másik felhasználót akinek „quest" jogosultsága van a tartományra. Továbbá mindegyik csak a saját intézményi munkaállomásaira tudjon bejelentkezni. Ezek lesznek a „Kérelmező" felhasználók. Bejelentkező nevük és jelszavuk lehet nyilvános az adott intézményen belül. A trükk csupán annyi, hogy a bejelentkezéskor kicseréljük a shell-t az IE-re, ami meghívja automatikusan a kitöltendő form-ot teljes képernyőn. Így nem lehet semmit rosszalkodni a kitöltésen túl a bejelentkezés alatt. Továbbá, ha bezárjuk ez IE-t, akkor az egyúttal a Windowsból való kijelentkezést is vonja maga után.
Az, hogy mindezt hogyan is kell megoldani, a következő szám cikkére marad. Ahogy az is, hogy miként küldjük a kérvényezés lezárásakor az automatikus e-mail üzeneteket a megfelelő vezetőnek jóváhagyásra, és hogyan kerülnek át a létrehozott „account"-ok az egyik OU-ból a másikba.
Addig is aki ennél többet szeretne tudni a témáról, az bátran keresse fel az IQSOFT - John Bryce Oktatóközpont munkatársait...

FARKAS VIKTOR
IQSOFT - John Bryce Oktatóközpont
farkas_v@hotmail.com
MCSE, MCT, HP-ASE

Dr. Watson

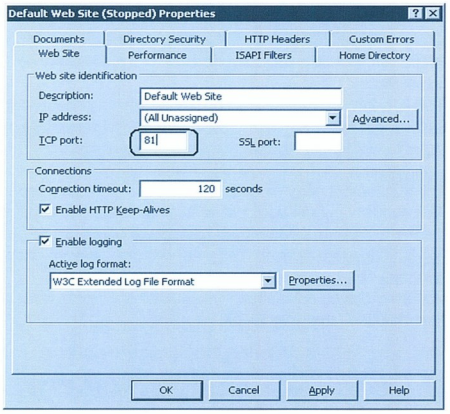
SSL-tanúsítványkérés ISA Server számára Avagy minek nekem IIS a tűzfalra?

Ebben a cikkben leírom, hogyan lehet megvalósítani egy külső hitelesítésszolgáltató által kibocsátott tanúsítvány segítségével a biztonságos (SSL) kapcsolatot az ISA Server 2004 és a külvilág között.

Lejárt az SSL-tanúsítványunk. Ez jó alkalomnak kínálkozott arra, hogy lépésről lépésre leírjam, hogyan is tesz szert az ember egy vadonatúj SSL-tanúsítványra mondjuk a NetLocktól. A kihívás abban rejlik, hogy a tanúsítványkérelmet fájlban kell benyújtani a NetLock felé, amit sem a Windowsban lévő Certificates MMC-konzol, sem az ISA Server nem tud létrehozni. Ellenben az IIS igen. Ha még eddig nem volt IIS fent az ISA Serveren, hát most – kényszerből, átmenetileg - fel kell rá telepíteni, és ott is kell hagyni mindaddig, amíg az egész folyamat le nem zajlik, vagyis körülbelül három hétig!

IIS telepítése és tanúsítványigénylés készítése

Miután feltettünk egy alap IIS-t (ennek lépéseivel nem untatnám a tisztelt olvasókat), át kell állítanunk, hogy melyik portot használja a Default Web Site, mivel a szokásos 80-as porton az ISA csücsül. Az alábbi ábrán látható, hogy hol lehet ezt megtenni (én a 81-es portot választottam a 80-as helyett). Amíg ezt meg nem tesszük, a Default Web Site-ot el sem lehet indítani!



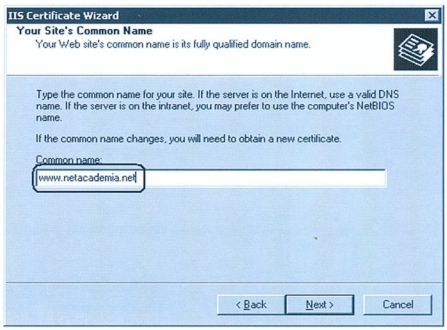
Átállítjuk a Default Web Site által használt portot

Ezután már elindítható a site. A következő lépés máris a tanúsítványkérés lesz. Menjünk vissza a Default Web Site tulajdonságlapjára, és válasszuk ki a Directory Security lapot! A „Server Certificate…" gombon kattintva elindul a tanúsítványvarázsló, melyben a „Create new certificate" <NEXT> „Prepare the request nowbut send it later" <NEXT> irányban haladva arra kérjük az IIS-t hogy ne keressen egy közelben lévő, online CA-servert, hanem a tanúsítványigénylésünket tegye félre szebb napokra.

A következő lapon a tanúsítvány becenevét (display name), az RSA-kulcs hosszát (mondjuk 1024 bit) és a CSP-t (Crypto Service Provider, a kulcsgeneráló és tároló komponens) állíthatjuk be, ezek közül a becenév az, ami mindenképpen módosításra szorul, tekintve hogy nincs kitöltve. Akinek nem elég az 1024 bit, itt avatkozzon be, aki pedig nem szeretné, hogy a kész RSA-kulcspár az All Users profilban tárolódjon, izzítsa be a Smart Cardját, és állítsa be CSP-nek azt. Click <NEXT>.

A következő oldalon az Organization mezőt a cégbírósági bejegyzésnek megfelelően kell kitölteni, ellenkező esetben a tanúsítvány kibocsátója nem fogadja el az igénylésünket! Click <NEXT>.

A következő lapot megint idemásolom, mert ez létfontosságú adatot tartalmaz: milyen néven fognak minket megtalálni a kuncsaftjaink a weben?

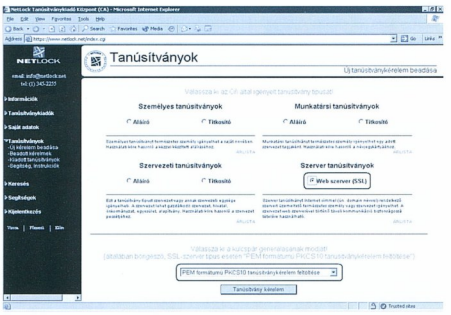


A DNS-zónába felvitt név megadása

Ha ezt a mezőt elrontjuk, a kész tanúsítvány ellenőrzésekor állandóan hibaüzenetet fogunk kapni, hogy a tanúsítványban szereplő név és a hivatkozott név eltér. Ebből egyébként sajnos az is következik, hogy az SSL-es webhelyeket csak és kizárólag az itt megadott néven érhetjük el hibamentesen, de például IP-címmel már nem! Rövid névvel szintén nem! Csak úgy, ahogy itt megadtuk, punktum.
Második fájdalmas következmény: ha egy gépen több különböző (nevű) site fut, sajnos nem tudnak osztozni a tanúsítványon, mindegyiknek saját SSL-tanúsítványra van szüksége – az eltérő név miatt. Click <NEXT>.
A következő lapon adjuk meg a településünk és megyénk nevét. Click <NEXT>. Click <NEXT>. Click <FINISH>.
Az eredmény a C:\CERTREQ.TXT fájlba került, ami nem más, mint egy PKCS10-es formátumú fájl (ennek később lesz némi jelentősége). Hát ez még nem a NetLock! Most további küzdelmek várnak ránk, ezúttal azonban nem az IIS-sel, hanem a szolgáltató webalkalmazásával kell megbirkóznunk.

Az igénylés benyújtása

Általában nem javasolható, hogy az ISA, vagy bármelyik publikált szerverünkről az internetet böngésszük, de most mégis ez következik. Noha magát az igénylést még elintézhetnénk egy másik gépről (átvisszük a CERTREQ.TXT-t), a kész tanúsítvány letöltését már mindenképpen itt kell elvégeznünk, hiszen az igénylés nem csak ebből a fájlból áll, hanem a máris legenerált RSA-kulcsokból is, akik itt, ezen a gépen várnak minket!
A következőkben – részrehajló módon – a NetLock Kft.-nél követendő lépéssorozatot mutatom meg, tulajdonképpen azért, mert úgysincs Magyarországon még egy hitelesítésszolgáltató, akinek a szervertanúsítványát bármelyik vadidegen weblátogató el tudná fogadni, tehát gyakorlatilag nincs konkurrencia, aki megsértődhetne. Esetleg a VeriSign, de nekik nem szólunk a cikkről.
A NetLock-ügyfélmenü létrehozásával ismét nem fárasztom Önöket, klikk, klikk, jelszó, klikk, ok és hasonlók után eljutunk a tanúsítványigénylési weblaphoz:



SSL-tanúsítvány igénylése PKCS10-es fájllal

Válasszuk ki a Szervertanúsítványok közül az ott árválkodó egyetlen árucikket, a Web szerver tanúsítványt. Nini! PKCS10-es formátumban várják a tanúsítványigénylést! Nekünk pont ilyenünk van! Nosza, igényeljünk tanúsítványt! A következő lapon fel kell vinnünk a webkiszolgálónk publikus nevét, mégpedig pontosan ugyanazt a nevet, amit korábban a tanúsítványkérelemben is megadtunk!
Most egy igen bonyolult lépés következik: a létrehozott szerver rekordján el kell találnunk az egér bal gombjával a jobbra mutató (bekarikázott) pici kék háromszöget! Ez valami intelligencia- vagy koncentrálóképesség-teszt lehet, aki erre nem képes, ne is álmodjon SSL-tanúsítványról!

/▶/ - Kérjük válasszon a szerver regisztrációk közül a kis kék nyíl segítségével!

Kiválasztjuk az új kiszolgálót a „többi" közül

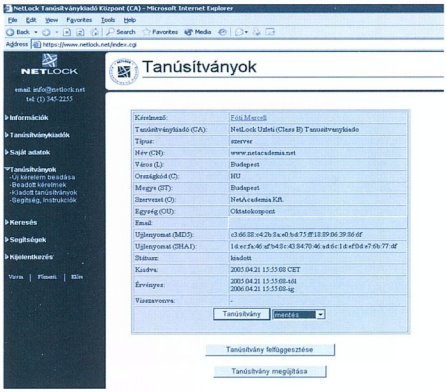
A következő ablakba be kell másolni a CERTREQ.TXT teljes tartalmát, ami valahogy így fog kinézni:

Feltöltjük a tanúsítványkérelmet (CERTREQ.TXT)

Ezután a tanúsítvány „erősségét" kell kiválasztanunk (A, B vagy C osztály), vállalati kiszolgálókhoz minimum a B osztály ajánlott, a fizetés, a belépési nyilatkozat és az egyéb iratok benyújtása, majd 3 hét szünet következik.

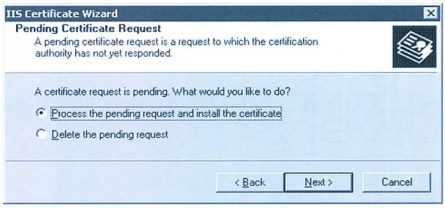
3 hét múlva

Miután kivártuk a szükséges ügyintézési-fizetési-időhúzási időszakot, értesítés kapunk a NetLocktól a tanúsítvány elkészültéről.
(Kis közjáték: szerencsétlen módon én ebben a periódusban egy hétig külföldön voltam, miközben idehaza kiderült, hogy a cégbírósági kivonat és egyéb papírok mellé kell még egy ingyombingyom is, amit egy hétig nem tudtam produkálni. Így a három hétből négy és fél lett, a meglévő, korábbi eredeti SSL-tanúsítványunk pedig szép csendben lejárt. Nem jött össze a „rolling update", egy hétig szünetelt az SSL a www.netacademia.net-en. Tanulság: aki hosszadalmas shoppingba kezd, menet közben ne utazzon sehova!)
Az értesítőlevélben található linken kattintva a kész tanúsítvány letöltési lapjára repülünk:



A kész tanúsítvány letöltése

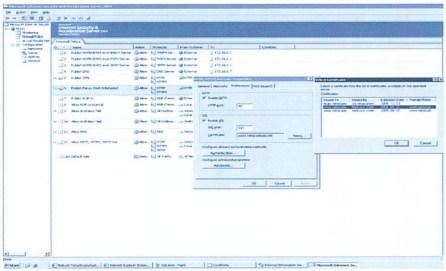
A mentés eredménye egy .CER kiterjesztésű x.509 típusú tanúsítvány lesz, benne a digitálisan aláírt publikus kulcsunkkal. Ezt most össze kell „pároztatni" a magánkulccsal, ami három hete erre az aktusra vár. (A magánkulcs természetesen nincs benne a letöltött .CER fájlban, mert ha benne lenne, megsértettük volna a PKI egyik legfontosabb szabályát: „magánkulcs soha nem mehet át a hálózaton". Ezért kell most segítenünk, hogy a kulcspár két része egymásra találjon.)
Nemcsak a magánkulcs, hanem az IIS is három hete erre vár, tehát ha elindulunk az SSL-varázslóval (lásd fent), ezalkalommal másik, kiéheztetett arcát mutatja:



A letöltött tanúsítvány „megetetése" az IIS-sel

Itt természetesen a „Process..." ágon haladunk tovább, mert a „Delete..." egész eddigi munkánkat hajítja ki, ami távolról sem lehet a célunk. Az IIS Certificate Wizard az összepárosított nyilvános- és magánkulcsot a számítógép tanúsítványtárába helyezi el, amit az ISA Server is elér. Utolsó lépésként az ISA Serverben létre kell hozni egy Web Listenert a 443-as portra, és meg kell neki mutatni, hogy melyik tanúsítványt használja. Erről is készítettem fotót, ezzel zárnom a cikket is. Van azonban még két fontos dolgunk, mielőtt kényelmesen hátradőlhetnénk:

1. Ha az IIS-t kizárólag azért telepítettük az ISA Serverre, hogy a tanúsítványkérésben segítségünkre legyen, most már szedjük le.
2. A kész tanúsítványt magánkulcsostól-mindenestől exportáljuk ki egy jelszóval védett .PFX fájlba, írjuk CD-re vagy flopira, és tegyük el biztos helyre. Akik szeretik a kalandot, rejtsék el szteganografikus módszerrel mondjuk egy képben vagy egy MP3-ban, esetleg EXE fájlban a PFX fájlt, mert ott biztosan senki nem fogja keresni, és ezt tegyék ki CD-re. Így még ha lába is kél a lemeznek, mindenki azt fogja hinni, hogy azon csak vacak zene, vagy unalmas ISA-screenshotok vannak. Mint pl. az alábbi:



Az ISA web listener megkapja a kulcsot

Föti Marcell
marcellf@netacademia.net
A szerző a NetAcademia vezető oktatója
MCSE, MCT, MCDBA, MZ/X

„Eredeti Windows – Valódi Előny" program

A Microsoft 2005 februárjában 25 országra, köztük Magyarországra is kiterjesztette a Windows Genuine Advantage (WGA) – magyar nevén „Eredeti Windows - Valódi Előny" – programot, melynek célja az illegális szoftverhasználat visszaszorítása. Magyarországon február óta 11 ezer felhasználó tesztelte Windows szoftverének eredetiségét.

A nem hivatalos forrásokból származó, legálisnak hitt illegális szoftverek megvásárlása évről évre több millió egyéni felhasználónak és vállalatnak okoz károkat, hiszen ezek gyakran biztonsági réseket vagy kártékony kódokat is tartalmaznak, vagy egyszerűen csak hiányosak. Az „Eredeti Windows – Valódi Előny" kezdeményezés keretében lehetőség nyílik a szoftverek eredetiségének ellenőrzésére. A tesztelésen túljutott jogtiszta Windowst használók számára a Microsoft ingyenes hozzáférést biztosít különböző népszerű programokhoz, illetve további előnyökhöz juttatja az egyéni és vállalati felhasználókat.

Az „Eredeti Windows – Valódi Előny" kezdeményezés sikerét jelzi, hogy az angol nyelvterületen elsőként elindított Központot alig öt hónap alatt 5 millió felhasználó kereste meg. Magyarországon május elejéig több mint 30 ezer felhasználó látogatott el a Letöltő Központon keresztül az „Eredeti Windows - Valódi Előny" tájékoztató honlapjára, közülük 11 ezren vették igénybe a program által biztosított eredetiség-vizsgálatot. A jogtiszta Windowst használók által leginkább kedvelt, ingyenesen letölthető alkalmazások a Windows Media Player 10 és a Photo Story 3 voltak.

A Microsoft, a legális Windowst használók számára, ezen felül olyan hasznos és fontos szoftverekhez biztosít ingyenes hozzáférést, mint például a Windows 98, Windows NT és Windows 2000 rendszerekkel is használható Media Player 9, a valutaváltóval kombinált számológép és a Windows Media kódoló 9-es sorozata.

2005 második felétől a kezdeményezés egy újabb fázisához érkezik. Ennek keretében a programban való részvétel kötelezővé válik azoknak, akik a Microsoft Letöltő Központ

`(http://www.microsoft.com/downloads/)`

szolgáltatásait igénybe kívánják venni. Ebben az időszakban az eredeti Windowst használók számos új alkalmazáshoz kapnak ingyenes hozzáférést a Letöltő Központon keresztül. A Windows Update

`(http://update.microsoft.ncom/)`

webhelyek biztonsági frissítései továbbra is mindenki számára elérhetőek maradnak.

Visual Studio 2005, SQL Server 2005 – új verziók

A Microsoft áprilisban a CTP ügyfelekhez (Community Technology Preview – Közösségi Technológiai Előzetes) eljuttatta a Visual Studio 2005 Beta 2, a Microsoft .NET Framework 2.0 Beta 2 és az SQL Server 2005 változatait, ezzel is jelezve, hogy elérkezett a fejlesztési ciklus utolsó szakaszához. A három termék együttese nagymértékben integrált fejlesztői és adatkezelő felületet biztosít. A Microsoft bejelentette a Microsoft Go-Live licencprogramot azon ügyfelek számára, akik azonnal telepíteni kívánják a Visual Studio 2005 és az SQL Server 2005 Express Edition segítségével készült alkalmazásokat.

Az ügyfelek visszajelzéseire reagálva a Microsoft kiegészítette a Microsoft .NET Framework 2.0 Beta 2 végfelhasználói licencszerződést (End User License Agreement – EULA), amelynek értelmében az ügyfelek a Beta 2 kiadás bázisán készült alkalmazásokat éles működési környezetben is használhatják. Ez a Go-Live licencnek nevezett függelék nem csak az ASP.NET webes alkalmazások bevezetését engedélyezi, hanem első ízben biztosít Go-Live licencet a WindowsForms, a Visual Studio Tools for Office-alapú alkalmazások, és a jelenlegi valamint a jövőben megjelenő Windows Mobile alapú eszközöket támogató .NET Compact Framework alkalmazások számára is. Professzionális fejlesztők ún. Visual Studio 2005 Beta Experience program keretében férhetnek hozzá a Beta 2 változatokhoz:

`http://www.microsoft.com/betaexperience.`

A .NET-el még csak most megismerkedni szándékozók, valamint a programozással hobbi szinten foglalkozók számára a Beta 2 kisebb, ún. Express változatai a legmegfelelőbbek. Ezek szabadon letölthetők a webről az alábbi linken:

`http://lab.msdn.microsoft.com/express/.`

A Go-Live programról további információk:

`http://msdn.microsoft.com/vs2005/golive`

Az SQL Server 2005 Express Edition használatba vételéről és a hozzá kapcsolódó Go-Live licencprogramról további információk az alábbi webhelyen találhatók:

`http://lab.msdn.microsoft.com/express/sql/default.aspx`

A Visual Studio 2005 és SQL Server 2005 erőforrások teljes listája megtalálható az alábbi weboldalon:

`http://lab.msdn.microsoft.com/vs2005/resources/learning/default.aspx`

MI MÁR LÁTJUK,

ahogy a következő NAGY ÖTLET megszületik.

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

Neked lehetőség. Nekünk kihívás.™

Your potential. Our passion.™

Microsoft®